NPS-OR-02-006

NAVAL POSTGRADUATE SCHOOL
Monterey, California

CAPITAL INVESTMENT PLANNING AID (CIPA)—AN OPTIMIZATION-BASED DECISION-SUPPORT TOOL TO PLAN PROCUREMENT AND RETIREMENT OF NAVAL PLATFORMS

by

Javier Salmeron
Robert F. Dell
Gerald G. Brown
Anton Rowe

September 2002

20021119 058

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE September 2002	3. REPORT TYPE AND DATES COVERED Technical Report

4. TITLE AND SUBTITLE
CAPITAL INVESTMENT PLANNING AID (CIPA)—AN OPTIMIZATION-BASED DECISION-SUPPORT TOOL TO PLAN PROCUREMENT AND RETIREMENT OF NAVAL PLATFORMS

5. FUNDING
MIPR NO. N6227102MPRJK50

6. AUTHOR(S)
Javier Salmeron, Robert F. Dell, Gerald G. Brown, and Anton Rowe

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Operations Research Department
Naval Postgraduate School
Monterey, CA 93943-5219

8. PERFORMING ORGANIZATION REPORT NUMBER
NPS-OR-02-006

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Chief of Naval Operations
Assessment Division (N81)
2000 Navy Pentagon
Washington, DC 20350-2000

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5660

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words.)
    Capital Investment Planning Aid (CIPA) is an optimization-based decision support system custom-built for the U.S. Navy to formulate complete force structure plans that include ship, submarine, and aircraft procurement and retirement schedules over a 30-year planning horizon. This is an important problem, representing a $1 trillion (2002 dollars) commitment by the Navy. Each candidate plan must respect annual budgets in several funding categories, while following Navy planning guidance such as keeping shipyards efficiently employed, limiting the average and/or maximum age of platforms, and meeting Integrated Warfare Architecture (IWAR) requirements. Without CIPA, such alternatives must be manually assembled—a slow, laborious, demanding task fraught with opportunities for clerical error. CIPA offers a graphical interface to organize planning data, accepts ad hoc manual guidance, optimally completes the missing details of any alternate scenario in a second or two, displays its recommendations and their consequences, and provides scenario cataloging and comparison tools. CIPA reduces to minutes the planning cycle from exigent question to exploratory scenarios to PowerPoint slides displaying results. This document describes the planning environment into which CIPA has been introduced, how CIPA works, and shows how CIPA is used.

14. SUBJECT TERMS
Military Capital Planning; Investment Planning; Modeling and Simulation; Optimization; Heuristics.

15. NUMBER OF PAGES
130

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

CAPITAL INVESTMENT PLANNING AID (CIPA)—AN OPTIMIZATION-BASED DECISION-SUPPORT TOOL TO PLAN PROCUREMENT AND RETIREMENT OF NAVAL PLATFORMS

by

Javier Salmeron**, Gerald Brown*, Robert Dell*, Anton Rowe**
Operations Research Department
Naval Postgraduate School
Monterey, CA 93943-5001

Abstract

Capital Investment Planning Aid (CIPA) is an optimization-based decision support system custom-built for the U.S. Navy to formulate complete force structure plans that include ship, submarine, and aircraft procurement and retirement schedules over a 30-year planning horizon. This is an important problem, representing a $1 trillion (2002 dollars) commitment by the Navy. Each candidate plan must respect annual budgets in several funding categories, while following Navy planning guidance such as keeping shipyards efficiently employed, limiting the average and/or maximum age of platforms, and meeting Integrated Warfare Architecture (IWAR) requirements. Without CIPA, such alternatives must be manually assembled—a slow, laborious, demanding task fraught with opportunities for clerical error. CIPA offers a graphical interface to organize planning data, accepts ad hoc manual guidance, optimally completes the missing details of any alternate scenario in a second or two, displays its recommendations and their consequences, and provides scenario cataloging and comparison tools. CIPA reduces to minutes the planning cycle from exigent question to exploratory scenarios to PowerPoint slides displaying results. This document describes the planning environment into which CIPA has been introduced, how CIPA works, and shows how CIPA is used.

1

Table of Contents

List of Figures

List of Tables

Executive Summary

The Navy's procurement and retirement planning is part of a complicated Department of Defense budget planning process. The U. S. Navy will spend more than $1 trillion (2002 dollars) over the next 30 years to procure ships, submarines, and aircraft to enable it to fulfill its missions.

Today, an attack submarine costs more than $2 billion, an aircraft carrier more than $5 billion, and its air wing $5 billion more. The Navy must balance these large capital expenditures with other procurements and maintain an industrial base capable of satisfying its unique requirements.

Capital Investment Planning Aid (CIPA) is a force structure planning tool that can be used to prescribe ship, submarine, and aircraft procurement and retirement schedules over a 30-year planning horizon. Without CIPA, plans must be manually assembled—a slow, laborious, and demanding task fraught with opportunities for clerical error, and limited to a small range of alternatives. CIPA augments manual planning with optimization, recommending the best (or nearly best) yearly force structure procurement and retirement plan based on industrial and budget constraints, as well as mission inventory and force mix requirements. CIPA is the only Navy decision support system that integrates aircraft and ship procurement decisions with fiscal, industrial, and mission requirements to render the best integrated long-term advice.

The primary components of CIPA are a Graphic User Interface (GUI) and a Solver module. The GUI incorporates user-friendly displays to allow a force-structure analyst to easily create and modify a plan, by accepting ad hoc manual guidance, simplifying the visualization and interpretation of results, and facilitating related tasks such as import or export data and results, and organizing planning data. The CIPA Solver is comprised of a fast, custom heuristic that solves a planning scenario in a few seconds, and an exact method that can provide a solution with a finer quantitative assessment of its quality.

The graphical interface to organize planning data accepts ad hoc manual guidance, optimally completes the missing details of any alternative scenario in a second or two, displays its recommendations and their consequences, and provides scenario cataloging and comparison tools. CIPA reduces to minutes the planning cycle from exigent question to exploratory scenarios to PowerPoint slides displaying results.

This document presents an overview of CIPA. We briefly describe the CIPA planning environment, present an integer-linear program at the heart of CIPA, discuss exact and heuristic techniques we employ to solve CIPA, along with their computational performance, and provide an overview of the graphical user interface.

1. Procurement and Retirement Planning for Navy Ships and Aircraft[1]

The Navy's procurement and retirement planning is only part of a complicated Department of Defense budget planning process. How did this process get so complex?

American defense budgeting began during the Revolution with proposed requisitions for fielding men and armaments, hand-written by the few well-known general officers who were preparing to personally lead these military operations. These requests were for "what I need." This requirements-based process persisted with some embellishment until after World War II, when the Hoover Commission required (1948) that budgets be defended in terms of function and activities, rather than just numbers of men and amounts of materiel. The Defense Department and its staffs asked for "what we need to be able to achieve these things, by these specific means." "In 1959, General Maxwell Taylor suggested a 'mission-oriented' budget... Congress subsequently asked that the budget for fiscal 1961 be based on 'functional categories.' The idea was to replace intermediate military 'inputs' by strategic 'outputs' directly describing the policy's intended effects..." [Martin 1988]. Subsequently, Secretary of Defense Robert McNamara introduced the five-year budget programs and a penchant for detailed decision-support that still characterizes defense budgeting. Now, we start with strategy, express this in terms of mission areas, and then eventually expand these into actual requirements for personnel, materiel, and, in particular, major weapons systems.

Naval spending has always involved large amounts of resources, research and technology, money, and the attention of civilian and military leadership. In 1794, President Washington asked Congress to authorize construction of six frigates at six different sites to help protect American merchant fleets from attacks by Algerian pirates and harassment by British and French forces [Hagan 1978]. With a total budget exceeding $800,000 (1794 dollars), congressional debate was intense, but construction was ultimately approved on the condition that it be conducted exactly as proposed in six different constituencies, thus affording political insulation. In fiscal year 1999 dollars, the frigates would cost $2.6 billion [Field 1999]. The USS Constitution (shown in Figure 1.1) employed revolutionary technology, used more than 1,500 trees felled from Maine to Georgia, and was armed with cannons cast in Rhode Island [USS CONSTITUTION 1999]. Today, an attack submarine costs more than $2 billion, an aircraft carrier more than $5 billion, and its air wing $5 billion more. These ships are the only current American clients for high-pressure steam nuclear power plants. The Navy must balance these large capital expenditures with other procurements and maintain an industrial base capable of satisfying its unique requirements.

Navy budget analysts must continually respond quickly to scenarios arising from emergent world events and domestic politics. Their advice must consider the complex interplay between past decisions, politics, and fiscal realities.

---

[1] This section relies substantially on text originally found in the first chapter of Field [1999] and the second Chapter of Garcia [2001].

The Navy's current effort to better manage this complex interplay is the Integrated Warfare Architecture (IWAR) Assessment and Planning Process.

1.1. IWAR Assessment and Planning

IWAR assessment and planning started in 1998 and is the responsibility of the Chief of Naval Operations Assessment Division (N81) [Chief of Naval Operations 2000]. There are five IWAR warfare components: Information Superiority/Sensors; Sea Dominance; Air Dominance; Power Projection; and Deterrence, and seven IWAR support components: Sustainment, Infrastructure, Manpower/Personnel, Readiness, Training/Education, Technology, and Force Structure. IWAR assessment and planning provide end-to-end capability analysis of naval forces that link warfare and support components.

The IWAR Force Structure component focuses "on assisting Navy leadership in best matching available resources with desired capabilities in the near, middle, and far terms" [Chief of Naval Operations 2000]. More specifically, the Force Structure component develops and analyzes alternate procurement and retirement plans for ships, submarines and aircraft that meet fiscal constraints [Valentine 1999]. One of the primary objectives is to quantify, in terms of dollars and capabilities, the effect of Ship Conversion Navy (SCN) and Aircraft Procurement Navy (APN) programs.
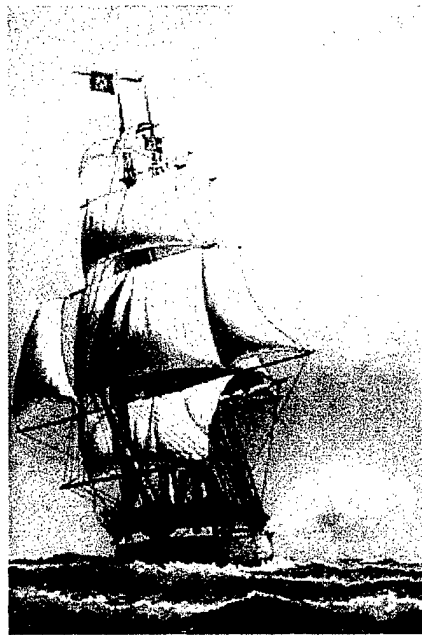
Figure 1.1: The USS Constitution exhibited innovative naval architecture and the latest armament technology. Figure from [All Hands 1997]. Construction of the Constitution was planned and approved in 1794 by the highest levels of American government, and required wide mobilization of resources.

1.2. EPA/TOA

Extended Planning Annex/Total Obligated Authority (EPA/TOA) is the primary tool used by N81 to evaluate specific alternate force structures. Based on input from the warfare IWAR components, resource sponsors, and numerous documented requirements such as the Quadrennial Defense Review (QDR), Defense Planning Guidance (DPG), and Commander in Chief operational plans, analysts perform manual "what-if" scenarios using EPA/TOA. Analysts then compare scenario results to determine the structure that most closely matches projected budgets and meets force size and capability requirements.

Systems Planning and Analysis, Incorporated maintains EPA/TOA for N81. EPA/TOA consists of 62 spreadsheets (Figure 1.2) that calculate yearly Military Personnel (MILPERS), Civilian Personnel (CIVPERS), Military Pay Navy (MPN), Operation and Maintenance (O&M), Other Procurement Navy (OPN), Ship Conversion Navy (SCN), Aircraft Procurement Navy (APN), Procurement of Ammunition Navy/Marine Corps (PANMC), Weapon Procurement Navy (WPN), Research Development Technology & Experimentation (RDT&E), Military Construction (MILCON), Family Housing Navy (FHN), National Defense Sea-lift Fund (NDSF), and OTHER monies for input procurements and retirements.
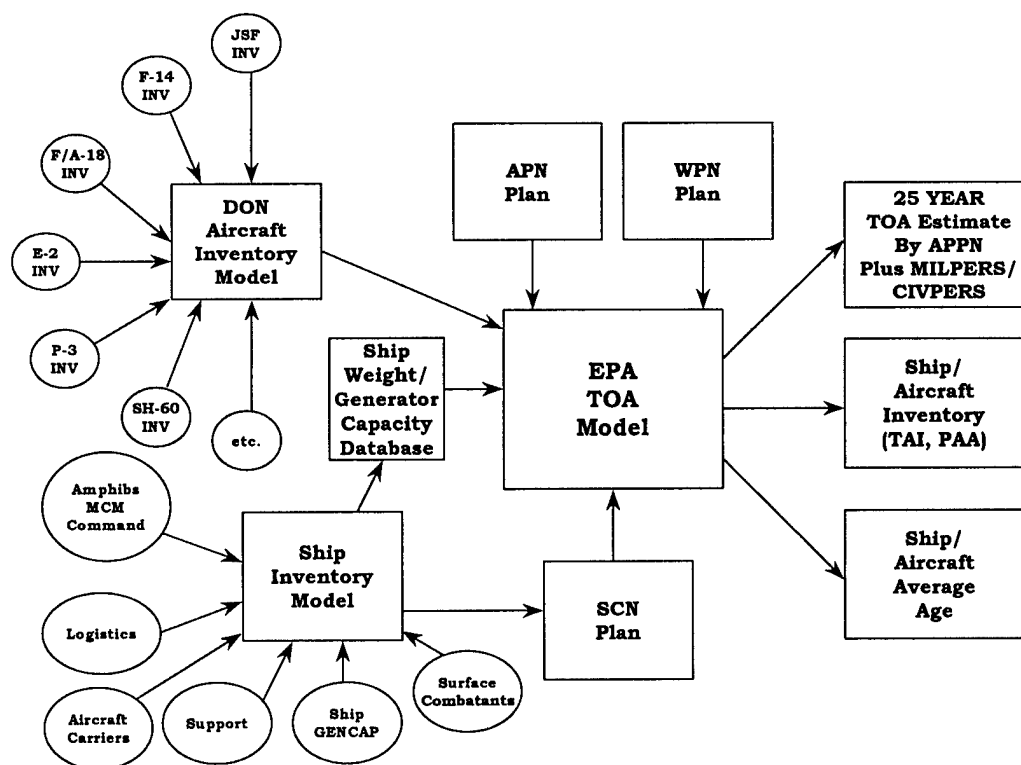
Figure 1.2: Extended Planning Annex/Total Obligated Authority (EPA/TOA) [Systems Planning and Analysis 1998]. EPA/TOA consists of 62 spreadsheets that are linked to estimate Total Obligated Authority.

The current Resource Allocation Display (RAD) in EPA/TOA—a snapshot of the Fiscal Year's Defense Plan (FYDP) at a specific point in time—is the basis for near-term cost, procurement, and retirement of weapon systems. EPA/TOA fixes TOA in the near term based on the FYDP. For the middle term and far term the analyst manually provides procurements and retirements of weapon systems. EPA/TOA calculates TOA based on cost estimation relationships for the categories of MILPERS, CIVPERS, MPN, O&M, OPN, SCN, APN, PANMC, and WPN monies. The model uses cost analogies—the multiplication of a historic data point by a scalar—to estimate cost for RDT&E, MILCON, FHN, NDSF, and OTHER monies.

Force structure analysts are primarily concerned with the procurement and retirement of ships, submarines, and aircraft. Ships are procured with SCN money and aircraft with APN money. Within EPA/TOA, procurement of ships and aircraft directly affect SCN and APN, and indirectly affect some of the other TOA monies through their respective cost estimation relationships.

Using EPA/TOA is labor intensive and error-prone. For instance, to change the procurement plan for the DDG51 class ship requires an analyst to make synchronous changes to three different spreadsheets, and this is just one of 100 platforms over a 25-year horizon. Each alternative accounts for numerous platform retirements and, recently, the 14 major procurement programs in process or under consideration.

1.3. Changing Force Structure Priorities

N81 planners face many problems determining and dealing with force structure priorities. Priorities change for many reasons including: a new President and administration, world events, and new technologies and systems. CIPA can help address some of the competing priorities and allow planners to quickly explore optimized alternatives in their ever-changing environment. Below we provide some recent examples of scenarios that typify those that must be considered by N81 Force Structure Planners.

The DPG outlines the missions the U.S. military must fulfill to satisfy U.S. National Military Strategy. The George W. Bush administration's plan for sizing the force structure started with a pledge to put strategic priorities first and budget priorities second [Scarborough 2001a]. President Bush directed Defense Secretary Rumsfeld to conduct a total review of the 1.36 million-person armed forces and reorganize it to meet the 21st century's threats. President Bush told our troops, "We must put strategy first, then spending. Our defense vision will drive our defense budget; not the other way around." [Scarborough 2001a]. Secretary Rumsfeld requested a $329 billion budget for 2002, which was the largest one-year defense increase since the 1980s. He implied that the 2002 budget is still considered to have far less funding than required to meet existing National Military Strategy. Secretary Rumsfeld also argued that the armed forces have been so under-funded and overused in the 1990s that one budget cycle cannot repair all the damage [Scarborough 2001a].

Secreatry Rumsfeld stated that the average age of aircraft has gone up about 10 years since the 1990s, and high maintenance costs are consuming the budget [Thomas 2001].

11

The Navy is forced to invest valuable maintenance man-hours on aircraft cannibalization, transferring scarce parts from aircraft to aircraft. He also stated that the "ship-building budget at the current rate is on a trajectory from 310 ships to 230 ships" [Thomas 2001]. The Bush administration's challenge is persuading Congress to supply the money necessary to rejuvenate the aging fleet.

The initial 2001 QDR stated that U.S. forces must be sized and shaped to perform three major tasks concurrently: defend the U. S. against attacks on the homeland or on defense-related information infrastructure; deter forward in critical areas of the world; and win decisively against an adversary in any one of these critical areas of the world [Grossman 2001]. Secretary Rumsfeld later revised the QDR to eliminate the requirement to perform the major tasks concurrently. This change to QDR guidance reflects the compromises being made to fulfill mission requirements while meeting tight budget realities. Defense planners acknowledge that the mismatch between strategy and resources has created a large number of budget shortfalls. One of these is military modernization. The military wants to get away from having aircraft, ships, and other equipment that are extremely old and drive up operating and maintenance costs [Weinberger 2001].

World events impact our Defense budget and force structure planning. The USS Cole attack [Navy Public Affairs Library, 2000] and the EP-3 collision with a Chinese fighter [Navy Public Affairs Library, 2001] are recent examples with minimal initial impact on naval inventories, but with potential widespread influence on force structure planning.

On 11 September 2001, terrorists crashed two hijacked commercial airliners into the twin towers of the World Trade Center in New York City and a third jet into the Pentagon [Rhem 2001]. In the wake of these terrorist attacks, Congress approved $40 billion in emergency defense funds. The Pentagon plans to spend half of the first $2.5 billion installment on intelligence upgrades and is expected to spend an additional $1 billion with the next installment [Capaccio 2001]. The Pentagon plan is to improve intelligence surveillance and reconnaissance aircraft, to buy more unmanned reconnaissance planes and private-source satellite imagery, and to upgrade the Pentagon's aging fleets of surveillance and tanker aircraft. The Navy is also considering accelerating purchases of C-40 transport planes to replace its much older C-9 cargo planes [Pasztor et al. 2001].

Since President Bush declared war on terrorism, more money has been promised to the Defense Department. The QDR retains 12 Navy carriers [Scarborough 2001b]. The big question is whether more money will be available to upgrade the rest of the fleet. Anti-terrorist operations will place more wear and tear on a combat fleet that already needs updated platforms. Another question is what additional money will be provided to pay for operating and maintaining the Navy's ships and planes already deployed in support of the war on terrorism.

New technologies and systems change the way we perceive and react to threats. These altered perceptions serve to shape our National Military Strategy, the Defense Planning Guidance, and consequently, our force structure planning. The tri-service,

multi-national Joint Strike Fighter (JSF) program (Figure 1.3), V-22 Osprey, Unmanned Combat Air Vehicle (UCAV), and Unmanned Air Vehicle (UAV), are examples of aircraft that will impact our force structure for the next decade and beyond.

The Marine Corps will get $592.3 million less than requested and build nine (instead of 12) V-22 Osprey tilt-rotor aircraft next year under the new defense bill approved by the Senate Armed Services Committee [Whittle 2001]. A special Pentagon panel recommended that Osprey production be held to a minimum while flaws that led to one of last year's crashes are fixed. The Marine Corps wants 360 V-22s to replace Vietnam-era helicopters.

New systems such as the Predator UAV are being used to support intelligence, surveillance, and reconnaissance missions while minimizing risk to our pilots and aircrew. The UCAV in Figure 1.4 is the next step toward minimizing combat fatalities while supporting two major combat roles: Suppression of Enemy Air Defenses (SEAD) and precision strike. The initial operational capability of UCAV is now planned for approximately 2010 [Baker 2001].

The multi-billion dollar JSF, V-22, UAV, and UCAV programs may affect our defense budget for decades, and significantly alter the way we prepare for and fight future battles. Force structure planners require flexible tools to deal with new system capabilities, uncertainties, and vulnerabilities.

Figure 1.3: An artist's rendition of the Lockheed Martin Joint Strike Fighter (JSF). The procurement plan calls for the Navy to buy 480 carrier versions and 609 Marine Corps short take-off and vertical landing (STOVL) versions. The $200 billion JSF contract is the largest in U.S. military history. Figure from [LockheedMartin.com, 2001].

Figure 1.4: An artist's rendition of the Boeing Unmanned Combat Air Vehicle (UCAV). The UCAV is the next step toward minimizing combat fatalities, while supporting two major combat roles: suppression of enemy air defenses and precision strike. Figure from [Boeing.com, 2001].

2. CIPA Components

The primary CIPA components are a Graphic User Interface (GUI) and a Solver module (CIPA Solver).

The GUI is implemented in Microsoft Excel [2002]. It incorporates user-friendly displays (including tables, graphics, reports, etc.) to allow a force-structure analyst to easily create and modify a plan, view and interpret results, and import or export data and results.

The CIPA Solver consists of two components:

- a fast, custom heuristic that solves a planning scenario in a few seconds, and

- an optional exact method that can provide a solution with a quantitative assessment of its quality. The exact approach requires the use of additional commercial off-the-shelf software (e.g., GAMS [Brooke et al., 1998]).

Figure 2.1 depicts typical CIPA use: The planner provides scenario data and guidance using the GUI (in some combination of manual data entry and retrieval from external databases); the solver is invoked (either the fast heuristic or a combination of fast heuristic and exact solver); and the solution is sent back to the GUI.
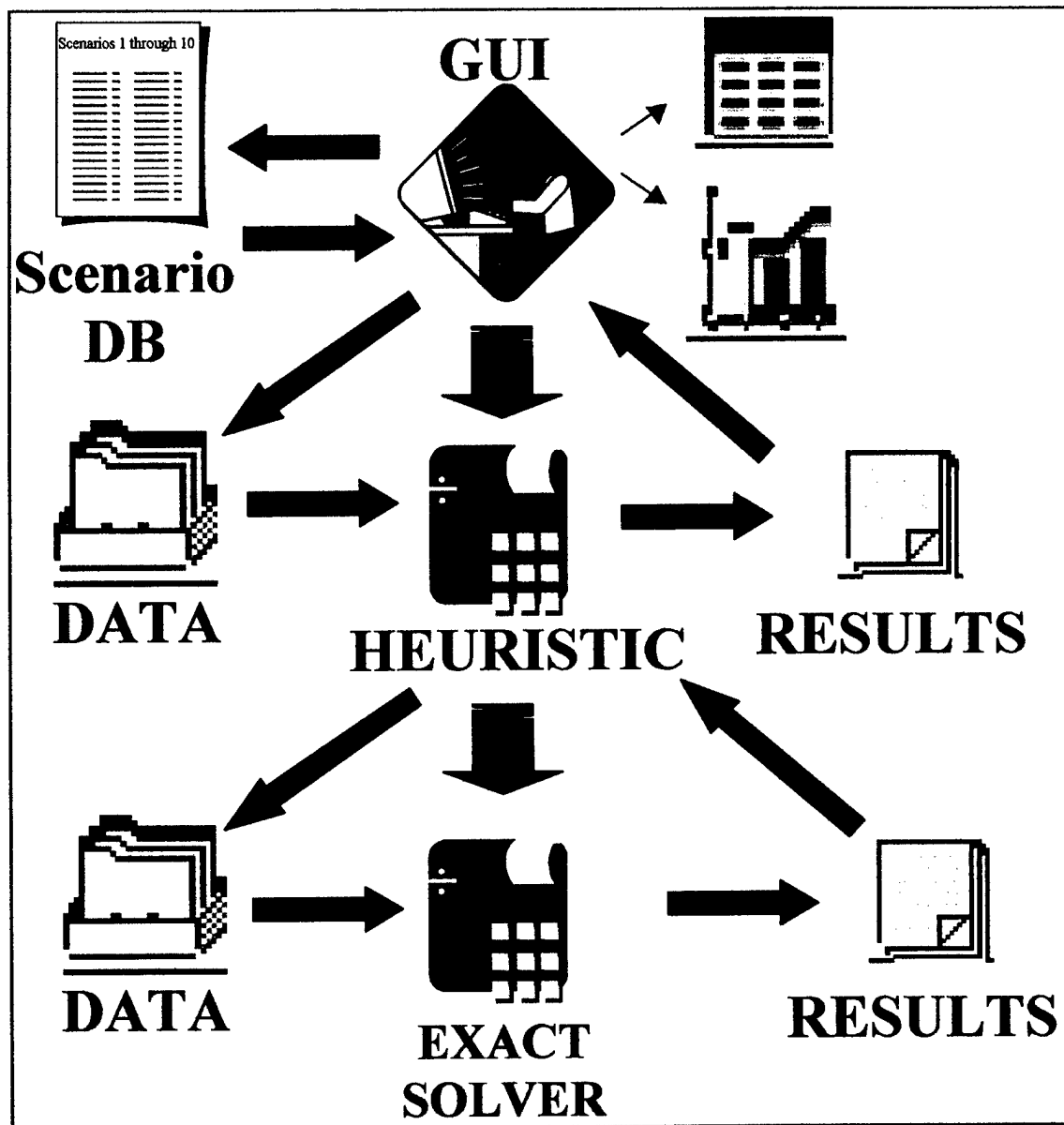
Figure 2.1: CIPA scheme. The planner provides scenario data and guidance using the GUI (in some combination of manual data entry and retrieval from external databases); the solver is invoked (either the fast heuristic or a combination of fast heuristic and exact solver); and the solution is sent back to the GUI.

3. CIPA Integer Linear Program

3.1 Mathematical Model Overview

CIPA expresses each planning scenario as an integer-linear program minimizing penalties associated with violating budget constraints, production constraints, and inventory requirements. For a recommended plan, CIPA illuminates the required budget, purchase dates and quantities, production facility employment levels, and force levels. CIPA also isolates force level deficiencies inflicted by budget restrictions on procurements, production that cannot keep pace with procurement requirements, or for lack of any existing replacement for retired platforms. CIPA maintains yearly time fidelity for 25 or 30 years. Because it can take up to five years to build platforms, CIPA's prescriptions for the last few years of the planning horizon may suffer from end effects: The solution for the last years of the horizon may lack accuracy because no information for years beyond the horizon has been specified.

In short, the mathematical model represents the a number of features divided into six categories:

1. Mission:
   - Ship-mission and air-mission requirements

2. Inventory:
   - Initial inventory of ships and aircraft
   - Ongoing (resident) production of ships and aircraft
   - Minimum and maximum annual production of ships and aircraft
   - Maximum total production of ships and aircraft
   - Maximum annual inventory of ships and aircraft
   - Minimum and maximum annual ship and aircraft retirement

3. Cost:
   - Ship and aircraft cost profile
   - Economy-of-scale for ship and aircraft procurement
   - Operation and maintenance costs for each ship and aircraft

4. Budget:
   - Minimum and maximum annual budget available
   - Minimum and maximum cumulative budget available

5. Industry:
   - Work-force profile for ship production
   - Minimum and maximum annual work-force levels for ship industry

6. Penalty:
   - Tradeoff among budget shortfall (or surplus), industry work-force shortfall (or surplus) and mission shortfall

Mission requirements (category 1) drive platform procurement. Category 2 features account for yearly platform inventory levels. These impose shipyard capacity, minimum retirement levels, the age of existing platforms, etc. Category 3 considers CIPA cost-related features. Procurement costs are typically incurred and spread out over a number of years before a platform is delivered. The cost of purchasing platforms exhibits economy of scale. Category 4 specifies annual and cumulative expenditures, and should not exceed or fall below their respective specified limits. Category 5 refers to work-force requirements for ship production that are spread out over the production period of a ship. Ideally, workforce levels should stay within specified limits to prevent loss of industrial capability and to avoid overtime costs. The last category refers to CIPA penalty charges for each individual violation of budget, industry, or mission-required levels. The penalties express the tradeoff among the different shortfalls and surpluses in order to prioritize the satisfaction of those conditions deemed more critical by the user.

As main decision variables, we consider the number of platforms procured and retired every year. We add additional variables to specify the piece-wise linear approximation of non-convex cost associated with economies-of-scale. We also incorporate "elastic" variables to account for budget, industry, and mission requirement violations. The objective function expresses the sum of these violations. See Field [1999] for a discussion of how to select penalty values.

All these features are mathematically represented through the following linear program:

$$\begin{cases} \text{CIPA:} & \min \ F \\ & \text{s.t.} \quad (3.1) \text{ to } (3.46) \end{cases}$$

where the objective function, F, and the constraints (3.1) and (3.46), are described in detail in the following section.

3.2 CIPA Model

This section presents the mathematical formulation of the CIPA model.

3.2.1 Sets and Indices

- Time

Y,   set of years of the planning horizon; $y, y' \in Y$. For convenience it is assumed that $Y = \{1, 2, 3, ..., |Y|\}$.

- Platform

A,   set of aircraft types; $a \in A$
S,   set of ship classes; $s \in S$

- Mission

M^A, set of air missions; $m \in M^A$

M^S, set of Ship-Missions; $m \in M^S$

$A_m \subseteq A$, subset of aircraft types that perform mission $m \in M^A$

$S_m \subseteq S$, subset of ship classes that perform mission $m \in M^S$

- Production

I_a, set of cost increments for aircraft $a \in A$; $i \in I_a$

P, set of production facilities; $p \in P$

$P_s \subseteq P$, subset of facilities that produce ship class $s \in S$

Q_{spy}, set of quantities available for ship $s \in S$ procurement at facility $p \in P_s$ in year $y \in Y$. This set is defined in terms of the $\underline{sproc}_{spy}$ and $\overline{sproc}_{spy}$ parameters (see below) as follows: $q \in Q_{spy} = \{\underline{sproc}_{spy}, \underline{sproc}_{spy} + 1, \cdots, \overline{sproc}_{spy}\}$.

- Others

Z^+, set of non-negative integers, $Z^+ = \{0,1,2,...\}$

3.2.2 Parameters (and Units)

- Conventions

The word "procurement" or "to procure" refers to "delivery" or "to deliver," respectively, unless explicitly stated otherwise. Therefore, we refer to "procure" as the action that takes place at the moment (year) that the platform is delivered and available for use from that year onwards, regardless of when the real "procurement" arrangements were made.

The words "time period" and "year" will be used interchangeably.

The words "shipyard," "facility," and "plant" will be used interchangeably.

- Objective-related parameters: Penalties

$ampen_m$, penalty for shortage in completing Air-Mission $m \in M^A$ (\$ per aircraft)

$smpen_m$, penalty for shortage in completing Ship-Mission $m \in M^S$ (\$ per ship)

$bpen_y^+$, penalty for budget excess (\$ per \$)

$bpen_y^-$, penalty for budget shortage (\$ per \$)

$cbpen_y^+$, penalty for cumulative expenses excess (\$ per \$)

$cbpen_y^-$, penalty for cumulative expenses shortage (\$ per \$)

$lpen_p^+,$ penalty for labor excess at plant $p \in P$ (\$ per worker)

$lpen_p^-,$ penalty for labor shortage at plant $p \in P$ (\$ per worker)

- Constraint-related parameters: Used for index dependencies

$SBb_{sp},$ number of years before (starting at 0) the procurement of ship class $s \in S$ from plant $p \in P_s$ requires budget (i.e., in 0,1,... $SBb_{sp} - 1$ years before)

$SCb_{sp},$ number of years before (starting at 0) the procurement of ship class $s \in S$ from plant $p \in P_s$ requires labor (i.e., in 0,1,... $SCb_{sp} - 1$ years before) .

$SBa_{sp},$ number of years after (starting at 1) the procurement of ship class $s \in S$ from plant $p \in P_s$ requires budget (i.e., in 0,1,... SBa_{sp} years before)

$SCa_{sp},$ number of years after (starting at 1) the procurement of ship class $s \in S$ from plant $p \in P_s$ requires labor (i.e., in 0,1,... SCa_{sp} years before)

$ABb_a,$ number of years before the procurement of aircraft type $a \in A$ in which the aircraft is paid (at once)

- Constraint-related parameters: Ships

$sinv_s^0,$ initial inventory of class $s \in S$ ships (# ships)

$csproc_{sy},$ committed procurement of class $s \in S$ ships in year $y \in Y$ due to production in progress (# ships)

$\overline{sinv}_s,$ maximum number of class $s \in S$ ships in inventory (# ships)

$\overline{stot}_{sp},$ maximum number of class $s \in S$ ships to procure from plant $p \in P_s$ (# ships)

$\underline{sproc}_{spy},$ minimum number of class $s \in S$ ships to procure from plant $p \in P_s$ in time period $y \in Y$ (# ships)

Note: $\underline{sproc}_{spy} = 0, \forall s \in S, p \in P_s; \forall y \leq \max\{SBb_{sp}, SCb_{sp}\} - 1$ and

$\underline{sproc}_{spy} = 0, \forall s \in S, p \in P_s; \forall y \geq |Y| + 1 - \max\{SBa_{sp}, SCa_{sp}\}$ is required

$\overline{sproc}_{spy},$ maximum number of class $s \in S$ ships to procure from plant $p \in P_s$ in time period $y \in Y$ (# ships)

Note: $\overline{sproc}_{spy} = 0, \forall s \in S, p \in P_s; \forall y \leq \max\{SBb_{sp}, SCb_{sp}\} - 1$ and

$\overline{sproc}_{spy} = 0, \forall s \in S, p \in P_s; \forall y \geq |Y| + 1 - \max\{SBa_{sp}, SCa_{sp}\}$ is required.

- Constraint-related parameters: Aircraft

$ainv_a^0$,   initial inventory of type $a \in A$ aircraft (# aircraft)

$caproc_{ay}$,   committed procurement of type $a \in A$ aircraft in year $y \in Y$ due to production in progress (# aircraft)

$\overline{ainv}_a$,   maximum number of type $a \in A$ aircraft in inventory (# aircraft)

$\overline{atot}_a$,   maximum number of type $a \in A$ aircraft to procure (# aircraft)

$\underline{aproc}_{ay}$,   minimum number of type $a \in A$ aircraft to procure in time period $y \in Y$ (# ships)

$\overline{aproc}_{ay}$,   maximum number of type $a \in A$ aircraft to procure in time period $y \in Y$ (# ships)

$\underline{inc}_{ayi}$,   increment $i \in I_a$ lower bound for the number of type $a \in A$ aircraft to be procured in year $y \in Y$ (# aircraft)

$\overline{inc}_{ayi}$,   increment $i \in I_a$ upper bound for the number of type $a \in A$ aircraft to be procured in year $y \in Y$ (# aircraft)

$squad_a$,   squadron size for aircraft $a \in A$ procurement (# aircraft)

- Constraint-related parameters: Retirements

$\underline{csret}_{sy}$,   minimum cumulative number of class $s \in S$ ships to retire by the end of time period $y \in Y$ (# ships)

$\overline{csret}_{sy}$,   maximum cumulative number of class $s \in S$ ships to retire by the end of time period $y \in Y$ (# ships)

$\underline{sret}_{sy}$,   minimum number of class $s \in S$ ships to retire by the end of time period $y \in Y$ (# ships)

$\overline{sret}_{sy}$,   maximum number of class $s \in S$ ships to retire by the end of time period $y \in Y$ (# ships)

$\underline{caret}_{ay}$,   minimum cumulative number of type $a \in A$ aircraft to retire by the end of time period $y \in Y$ (# aircraft)

$\overline{caret}_{ay}$,   maximum cumulative number of type $a \in A$ aircraft to retire by the end of time period $y \in Y$ (# aircraft)

$\underline{aret}_{sy}$,   minimum number of type $a \in A$ aircraft to retire by the end of time period $y \in Y$ (# aircraft)

$\overline{aret}_{sy}$,   maximum number of type $a \in A$ aircraft to retire by the end of time period $y \in Y$ (# aircraft)

- Constraint-related parameters: Mission inventory

$\underline{smreq}_{my}$,   number of ships required for Ship-Mission $m \in M^S$ in time period $y \in Y$
(# ships)

$\underline{amreq}_{my}$,   number of aircraft required for air mission $m \in M^A$ in time period $y \in Y$
(# aircraft)

- Constraint-related parameters: Budget

$oscn_y$,   fixed SCN cost in year $y \in Y$ ($)

$ocscn_y$,   fixed SCN cost in year $y \in Y$ for ships not considered ($)

$frac$,   historical fraction of total SCN cost for ship outfitting

$oapn_y$,   fixed APN cost in year $y \in Y$ ($)

$ocapn_y$,   fixed APN cost in year $y \in Y$ for aircraft not considered ($)

apn_s,   historical fraction of total APN categories 1 through 4 required for
categories 5 through 7

oom_y,   fixed O&M cost in year $y \in Y$ for maintenance not considered ($)

$scostb_{spql}$,   SCN cost incurred l years before q class-s ships are procured from plant p,
for $s \in S$, $p \in P_s$, $q \in \bigcup_{y \in Y} Q_{spy}$, $l = \{0,1,\cdots,SBb_{sp}-1\}$ ($)

$scosta_{spql}$,   SCN cost incurred l years after q class-s ships are procured from plant p,
for $s \in S$, $p \in P_s$, $q \in \bigcup_{y \in Y} Q_{spy}$, $l = \{1,\cdots,SBa_{sp}\}$ ($)

$aacost_{ayi}$,   increment $i \in I_a$ procurement cost for type $a \in A$ aircraft in year $y \in Y$
($ per aircraft)

$abcost_{ayi}$,   increment $i \in I_a$ fixed procurement cost (intercept) for type $a \in A$ aircraft
in year $y \in Y$ ($)

$omship_{sy}$,   O&M cost for class $s \in S$ ship in year $y \in Y$ ($ per ship)

$omair_{ay}$,   O&M cost for type $a \in A$ aircraft in year $y \in Y$ ($ per ship)

$csbudget_y$,   committed budget in year $y \in Y$ due to ship production in progress ($)

$\underline{toa}_y$,   TOA budget lower limit for year $y \in Y$ ($)

$\overline{toa}_y$,   TOA budget upper limit for year $y \in Y$ ($)

$\underline{ctoa}_y$,   TOA cumulative budget lower limit for year $y \in Y$ ($)

$\overline{ctoa}_y$,   TOA cumulative budget upper limit for year $y \in Y$ ($)

22

- Constraint-related parameters: Labor

$clabor_{py}$,   committed labor in year $y \in Y$ at plant $p \in P$ due to production in progress (# workers)

$sworkb_{spqn}$,   required labor n years before q class-s ships are procured from plant p, for

$$s \in S,\ p \in P_s,\ q \in \bigcup_{y \in Y} Q_{spy},\ n = \{0,1,\cdots,SCb_{sp} -1\} \ (\text{\# workers})$$

$sworka_{spqn}$,   required labor n years after q class-s ships are procured from plant p, for

$$s \in S,\ p \in P_s,\ q \in \bigcup_{y \in Y} Q_{spy},\ n = \{1,\cdots,SCa_{sp}\} \ (\text{\# workers})$$

$\underline{pcap}_{py}$,   minimum production capacity at plant $p \in P$ in time period $y \in Y$ (# workers)

$\overline{pcap}_{py}$,   maximum production capacity at plant $p \in P$ in time period $y \in Y$ (# workers)

3.2.3 Decision Variables (and Units)

- Variables related to objective function and to elastic constraints

F,   objective function value

α_{my}^{AM},   Air-Mission $m \in M^A$ shortage in year $y \in Y$ (# aircraft)

α_{my}^{SM},   Ship-Mission $m \in M^S$ shortage in year $y \in Y$ (# ships)

α_y^{B+},   budget excess in year $y \in Y$ ($)

α_y^{B-},   budget shortage in year $y \in Y$ ($)

α_y^{CB+},   cumulative budget excess in year $y \in Y$ ($)

α_y^{CB-},   cumulative budget shortage in year $y \in Y$ ($)

α_y^{L+},   labor excess in year $y \in Y$ (# workers)

α_y^{L-},   labor shortage in year $y \in Y$ (# workers)

- Main decision variables

$APROC_{ayi}$,   number of type $a \in A$ aircraft to procure at the start of year $y \in Y$ in cost increment $i \in I_a$ (# aircraft)

$ARET_{ay}$,   number of type $a \in A$ aircraft to retire by the end of year $y \in Y$ (# aircraft)

$SPROC_{spyq}$,   one if facility $p \in P$ is to deliver $q \in Q_{spy}$ class $s \in S$ ships at the start of year $y \in Y$, and zero otherwise (0-1 variable)

$SRET_{sy}$,   number of class $s \in S$ ships to retire by the end of year $y \in Y$ (# ships)

23

- Control decision variables

AP_{ayi},   one if aircraft $a \in A$ is procured at the start of year $y \in Y$ in cost increment $i \in I_a$, and zero otherwise (0-1 variable)

$AINV_{ay}$,   inventory of type $a \in A$ aircraft at the start of year $y \in Y$ (# aircraft)

$AMINV_{my}$,   inventory for air mission $m \in M^A$ at the start of year $y \in Y$ (# aircraft)

$SINV_{sy}$,   inventory of class $s \in S$ ships at the start of year $y \in Y$ (# ships)

$SMINV_{my}$,   inventory for Ship-Mission $m \in M^S$ at the start of year $y \in Y$ (# ships)

$SBUDGET_y$,   amount of SCN money to budget for year $y \in Y$ ($)

$ABUDGET_y$,   amount of APN money to budget for year $y \in Y$ ($)

$OMBUDGET_y$,   amount of O&M money to budget for year $y \in Y$ ($)

$BUDGET_y$,   total amount of money to budget for year $y \in Y$ ($)

$LABOR_{py}$,   amount of labor required in year $y \in Y$ at plant $p \in P$ (# workers)

3.2.4 Formulation

$$min\ F = \sum_{y \in Y} \sum_{m \in M^A} ampen_m\ \alpha_{my}^{AM} + \sum_{y \in Y} \sum_{m \in M^S} smpen_m\ \dot{\alpha}_{my}^{SM} +$$

$$\sum_{y \in Y} bpen_y^+\ \alpha_y^{B+} + \sum_{y \in Y} bpen_y^-\ \alpha_y^{B-} + \sum_{y \in Y} cbpen_y^+\ \alpha_y^{CB+} + \sum_{y \in Y} cbpen_y^-\ \alpha_y^{CB-} +$$

$$\sum_{y \in Y} \sum_{p \in P} lpen_p^+\ \alpha_{py}^{L+} + \sum_{y \in Y} \sum_{p \in P} lpen_p^-\ \alpha_{py}^{L-}$$

subject to:

- Ship

$$\sum_{q \in Q_{spy}} SPROC_{spyq} = 1, \qquad\qquad \forall s \in S, p \in P_s; \forall y \in Y \qquad (3.1)$$

$$SINV_{sy} = sinv_s^0 + \sum_{y' \in Y | y' \leq y} csproc_{sy'} + \sum_{p \in P_s} \sum_{y' \leq y} \sum_{q \in Q_{spy'}} q\ SPROC_{spy'q} - \sum_{y' \in Y | y' \leq y-1} SRET_{sy'},$$

$$\forall s \in S; \forall y \in Y \qquad (3.2)$$

$$\sum_{y \in Y} \sum_{q \in Q_{spy}} q\ SPROC_{spyq} \leq \overline{stot}_{sp}, \qquad\qquad \forall s \in S, p \in P_s \qquad (3.3)$$

- Aircraft

$$\sum_{i \in I_a} AP_{ayi} = 1, \qquad\qquad \forall a \in A; \forall y \in Y \qquad (3.4)$$

24

$$\underline{inc}_{ayi}\, AP_{ayi} \leq APROC_{ayi} \leq \overline{inc}_{ayi}\, AP_{ayi}, \qquad \forall a \in A, i \in I_a; \forall y \in Y \qquad (3.5)$$

$$\underline{aproc}_{ay} \leq \sum_{i \in I_a} APROC_{ayi} \leq \overline{aproc}_{ay}, \qquad \forall a \in A; \forall y \in Y \qquad (3.6)$$

$$AINV_{ay} = ainv_a^0 + \sum_{y' \in Y | y' \leq y} caproc_{ay'} + \sum_{y' \in Y | y' \leq y} \sum_{i \in I_a} APROC_{ay'i} - \sum_{y' \in Y | y' \leq y-1} ARET_{ay'},$$
$$\forall a \in A; \forall y \in Y \qquad (3.7)$$

$$\sum_{y \in Y} \sum_{i \in I_a} APROC_{ayi} \leq \overline{atot}_a, \qquad \forall a \in A \qquad (3.8)$$

- Retirements

$$\underline{csret}_{sy} \leq \sum_{y' \in Y | y' \leq y} SRET_{sy'} \leq \overline{csret}_{sy}, \qquad \forall s \in S; \forall y \in Y \qquad (3.9)$$

$$\underline{caret}_{ay} \leq \sum_{y' \in Y | y' \leq y} ARET_{ay'} \leq \overline{caret}_{ay}, \qquad \forall a \in A; \forall y \in Y \qquad (3.10)$$

- Mission Inventory

$$SMINV_{my} = \sum_{s \in S_m} SINV_{sy}, \qquad \forall m \in M^s; \forall y \in Y \qquad (3.11)$$

$$SMINV_{my} + \alpha_{my}^{SM} \geq \underline{smreq}_{my}, \qquad \forall m \in M^s; \forall y \in Y \qquad (3.12)$$

$$AMINV_{my} = \sum_{a \in A_m} AINV_{ay}, \qquad \forall m \in M^A; \forall y \in Y \qquad (3.13)$$

$$AMINV_{my} + \alpha_{my}^{AM} \geq \underline{amreq}_{my}, \qquad \forall m \in M^A; \forall y \in Y \qquad (3.14)$$

- Budget

$$SBUDGET_y = oscn_y + (1 + frac)\Big(ocscn_y + csbudget_y +$$
$$\sum_{s \in S} \sum_{p \in P_s} \sum_{\substack{y' \in Y | \\ y \leq y' \leq y + SBb_{sp}}} \sum_{q \in Q_{spy'}} scostb_{spq,y'-y}\, SPROC_{spy'q} +$$
$$\sum_{s \in S} \sum_{p \in P_s} \sum_{\substack{y' \in Y | \\ y - SBa_{sp} \leq y' \leq y-1}} \sum_{q \in Q_{spy'}} scosta_{spq,y-y'}\, SPROC_{spy'q} \Big),$$

25

$$\forall y \in Y \qquad (3.15)$$

$$
ABUDGET_y = oapn_y + (1+apn_5)\Big(ocapn_y +
$$

$$
\sum_{a\in A}\sum_{i\in I_a} (aacost_{a,y+ABb_a,i}\ \ APROC_{a,y+ABb_a,i} +
$$

$$
abcost_{a,y+ABb_a,i}\ \ AP_{a,y+ABb_a,i})\Big),
$$

$$\forall y \in Y \qquad (3.16)$$

$$
OMBUDGET_y = oom_y + \sum_{s\in S} omship_{sy}\, SINV_{sy} + \sum_{a\in A} omair_{ay}\, AINV_{ay}, \quad \forall y \in Y \qquad (3.17)
$$

$$
BUDGET_y = SBUDGET_y + ABUDGET_y + OMBUDGET_y,
$$

$$\forall y \in Y \qquad (3.18)$$

$$
\underline{toa}_y \le \alpha_y^{B-} + BUDGET_y, \qquad\qquad\qquad \forall y \in Y \qquad (3.19)
$$

$$
BUDGET_y - \alpha_y^{B+} \le \overline{toa}_y, \qquad\qquad\qquad \forall y \in Y \qquad (3.20)
$$

$$
\underline{ctoa}_y \le \alpha_y^{CB-} + \sum_{y'\in Y|y'\le y} BUDGET_{y'}, \qquad \forall y \in Y \qquad (3.21)
$$

$$
\sum_{y'\in Y|y'\le y} BUDGET_{y'} - \alpha_y^{CB+} \le \overline{ctoa}_y, \qquad \forall y \in Y \qquad (3.22)
$$

- Industrial

$$
LABOR_{py} = clabor_{py} +
$$

$$
\sum_{s\in S|p\in P_s}\ \sum_{\substack{y'\in Y|\\ y\le y'\le y+SCb_{sp}}}\ \sum_{q\in Q_{spy'}} sworkb_{spq,y'-y}\, SPROC_{spy'q} +
$$

$$
\sum_{s\in S|p\in P_s}\ \sum_{\substack{y'\in Y|\\ y-SCa_{sp}\le y'\le y-1}}\ \sum_{q\in Q_{spy'}} sworka_{spq,y-y'}\, SPROC_{spy'q},
$$

$$\forall p \in P; \forall y \in Y \qquad (3.23)$$

$$
\underline{pcap}_{py} \le \alpha_{py}^{L-} + LABOR_{py}, \qquad\qquad \forall p \in P; \forall y \in Y \qquad (3.24)
$$

$$
LABOR_{py} - \alpha_{py}^{L+} \le \overline{pcap}_{py}, \qquad\qquad \forall p \in P; \forall y \in Y \qquad (3.25)
$$

- Non-negativity and bounds

$$
0 \le AINV_{ay} \le \overline{ainv}_a \qquad\qquad\qquad \forall a \in A; \forall y \in Y \qquad (3.26)
$$

26

$$AMINV_{my} \geq 0, \qquad\qquad \forall m \in M^A; \forall y \in Y \qquad (3.27)$$

$$0 \leq SINV_{sy} \leq \overline{sinv}_s, \qquad\qquad \forall s \in S; \forall y \in Y \qquad (3.28)$$

$$SMINV_{my} \geq 0, \qquad\qquad \forall m \in M^s; \forall y \in Y \qquad (3.29)$$

$$\underline{sret}_{sy} \leq SRET_{sy} \leq \overline{sret}_{sy}, \qquad\qquad \forall s \in S; \forall y \in Y \qquad (3.30)$$

$$\underline{aret}_{ay} \leq ARET_{ay} \leq \overline{aret}_{ay}, \qquad\qquad \forall a \in A; \forall y \in Y \qquad (3.31)$$

$$SBUDGET_y \geq 0, \qquad\qquad \forall y \in Y \qquad (3.32)$$

$$ABUDGET_y \geq 0, \qquad\qquad \forall y \in Y \qquad (3.33)$$

$$OMBUDGET_y \geq 0, \qquad\qquad \forall y \in Y \qquad (3.34)$$

$$BUDGET_y \geq 0, \qquad\qquad \forall y \in Y \qquad (3.35)$$

$$LABOR_{py} \geq 0, \qquad\qquad \forall p \in P; \forall y \in Y \qquad (3.36)$$

$$\boldsymbol{\alpha} \geq \boldsymbol{0} \qquad\qquad (3.37)$$

- Fixed variables

$$APROC_{ayi} = 0, \qquad\qquad \forall a \in A, i \in I_a; \forall y \in Y \mid y \leq ABb_a \qquad (3.38)$$

$$SPROC_{spy0} = 1, \qquad \forall s \in S,\ p \in P_s;\ \forall y \in Y \mid y \leq \max\{SBb_{sp},\ SCb_{sp}\} - 1 \qquad (3.39)$$

$$SPROC_{spy0} = 1, \qquad \forall s \in S, p \in P_s; \forall y \in Y \mid y \geq |Y| + 1 - \max\{SBa_{sp}, SCa_{sp}\} \qquad (3.40)$$

- Binary/Integer variables

$$APROC_{ayi} \in Z^+, \qquad\qquad \forall a \in A, i \in I_a; \forall y \in Y \qquad (3.41)$$

$$ARET_{ay} \in Z^+, \qquad\qquad \forall a \in A; \forall y \in Y \qquad (3.42)$$

$$AP_{ayi} \in \{0,1\}, \qquad\qquad \forall a \in A, i \in I_a; \forall y \in Y \qquad (3.43)$$

$$SPROC_{spyq} \in \{0,1\}, \qquad\qquad \forall s \in S, p \in P_s; \forall y \in Y; \forall q \in Q_{spy} \qquad (3.44)$$

$$SRET_{sy} \in Z^+, \qquad\qquad \forall s \in S; \forall y \in Y \qquad (3.45)$$

An additional constraint requires that:

$$APROC_{ayi} \text{ is a multiple of } squad_a, \ \forall a \in A, i \in I_a; \forall y \in Y \qquad (3.46)$$

Remark: This constraint is not explicitly stated in the formulation. However, notice that it can be easily addressed by setting the proper segment limits. For example, if $squad_a = 4$ then the segment limits could be:

$$\underline{inc}_{ay1} = 0 = \overline{inc}_{ay1}, \underline{inc}_{ay2} = 4 = \overline{inc}_{ay2}, \underline{inc}_{ay3} = 8 = \overline{inc}_{ay3}, \ldots$$

Notice that, unless $squad_a = 1$ (in which case there is no need for extra segments), the number of segments in the model is significantly increased.

3.2.5 Description of the Formulation

Specifically, the formulation serves the following purposes:

- The objective function, F, comprises the sum of all the penalties due to Air-Mission and Ship-Mission shortfall, budget deficit and surplus, cumulative budget deficit and surplus, and labor deficit and excess.

- Ship constraints (3.1) to (3.3) constrain ship procurement: (3.1) ensures that one option for ship procurement is executed yearly at each plant, (3.2) calculates the yearly ship inventory, and (3.3) limits the maximum procurement from each plant.

- (3.4) to (3.8) constrain aircraft procurement: (3.4) to (3.6) guarantee that procurements are made within the limits of one specific segment and without exceeding the general minimum and maximum. (3.7) calculates the yearly aircraft inventory and (3.8) limits the maximum total procurement throughout the years.

- Cumulative retirement goals are specified in (3.9) to (3.10).

- (3.11) to (3.14) keep track of platform inventory to perform each specific mission and then calculate mission shortfalls.

- Budget constraints (3.15) to (3.22) are as follows: (3.15) calculates the ship-budget per year, which depends on the payment profile for each specific ship that has been procured. (3.16) is the yearly aircraft budget, considering the segment cost definition. (3.17) determines O&M costs based on existing inventories. The total yearly budget is assessed in (3.18), which serves to compute deficits and surpluses on a yearly and cumulative basis in (3.19) to (3.22).

- Based on labor profiles for those ships that have been procured, we estimate the labor force level required at the different shipyards in equation (3.23). Then, we compute the lack of labor or excess in (3.24) to (3.25).

- (3.26) to (3.37) establish non-negativity and bounds for the decision variables. Among these bounds, there exist specified maxima and minima for platform inventory and retirement levels.

- Some variables need to be fixed in (3.38) to (3.39), since otherwise they would involve actions beyond the horizon limits.

28

- (3.41) to (3.45) specify those variables that need to be considered integer or binary. This also implies the integrality of other variables such as platform inventories and mission inventories.

- Finally, (3.46) requires the aircraft procurement to be a multiple of the squadron size. As the remark indicates, this can be accomplished by adding extra segments for those aircraft whose squadron size for procurement purposes is greater than one.

4. CIPA Graphical User Interface

This section presents an overview of the CIPA Graphical User Interface (GUI). For details on CIPA data structures and relationships between the GUI and the Solver, see Appendix A of this document. For details on the GUI, we refer to CIPA: User's Manual [2002].

The GUI is developed in Microsoft Excel [2002]. The Excel workbook interface is organized in a number of input and output screens, the optimization solver link, and report screens.

4.1 GUI Basics

Each screen in the GUI contains three main regions (Fig. 4.1):

- A *Main Menu*, available on every screen, is located on the left-hand side.
- A dark blue *Header Bar* identifies the information shown on the screen.
- A *Data Screen* displays the screen's data and/or graphics.
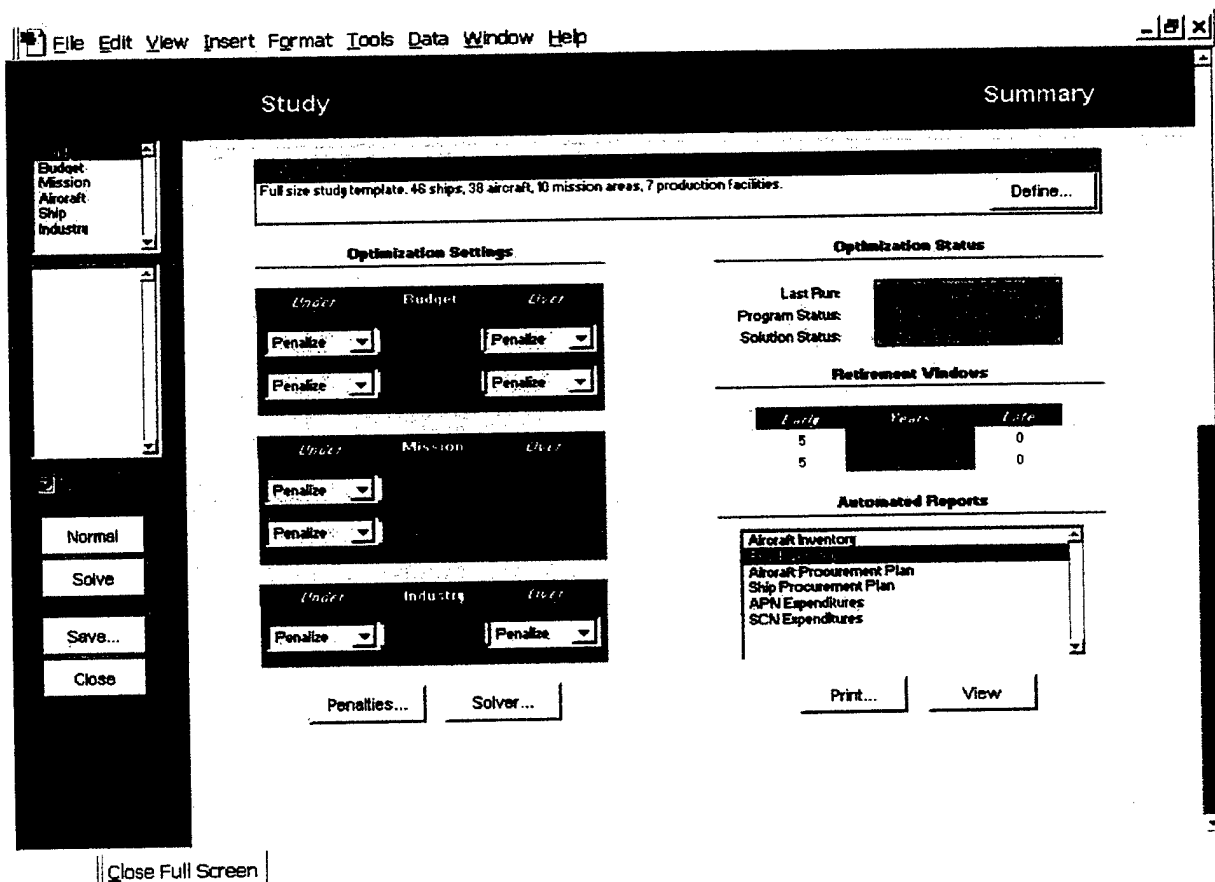
Figure 4.1: CIPA Workbook screen organization. The Main Menu (or Side Bar) appears on the left of every CIPA screen. The Header Bar on top identifies the contents of the Data Screen below it.

30

From the Main Menu located on the dark-gray bar on the left, we can:

- Access different data screens using the *Navigation* list boxes. For every screen item selected in the upper box, a subset of subordinate screen items appears in the lower box, and any of these subordinate Data Screens can be displayed.

- Optimize the plan by clicking the *Run* button. This invokes the Solver. Depending on the problem's complexity, user settings, and Solver request, the Solver may take just a few seconds, or hours. When the Solver finishes, the new solution is updated on the screen. (Note: Only one Solver run at any one time is permitted, even if several studies are open simultaneously.)

- Switch between a detailed data view and a graphic view using the *View Graphics* checkbox. This option makes it easier to visualize and understand the data and results.

- Toggle in and out of full-screen mode using the *Zoom* button.

- Save changes to the study by clicking the *Save* button. This supports analysis of multiple scenarios and keeps track of the impact that data changes have in the consequent optimal plans.

- Close the study and return Excel to normal mode by clicking the *Close* button.

4.2 Data Screens

We can use any Navigation list box to change the data screen viewed. If a screen has an associated graphic, checking View Graphics on the Main Menu will make this graphic visible. When View Graphics is unchecked, the underlying data is displayed instead.

CIPA GUI workbooks contain a variety of data screen types:

- Study Summary: General settings for the Solver and its status.

- Budget Summary: Yearly and cumulative budget (available and used).

- Budget Item: Yearly fixed and other cost by category (APN, SCN, O&M) (available and used).

- Mission Summary: Mission achievement relative to desired goal.

- Mission Element: Individual mission requirements (goal and achieved).

- Force Summary: Force components, categories, and retirement windows (input).

- Force Platform Aircraft procurement: Accounting of aircraft bought, retired, and retained, O&M rates, etc. (input data and Solver results).

- Force Platform Ship procurement: Accounting of ships bought, retired, and retained, O&M rates, etc. (input data and Solver results).

- Industry Summary: Aggregate annual labor usage for all shipyards and plants (Solver results).

- Industry Facility: Minimum and maximum annual labor usage for each shipyard or plant (input data).

The graphic and data elements contained in each of these data screens are shown in Figures 4.2 through 4.16. A detailed description of functionality can be found in CIPA: User's Manual [2002].

Budget   Summary

Total Obligation Authority
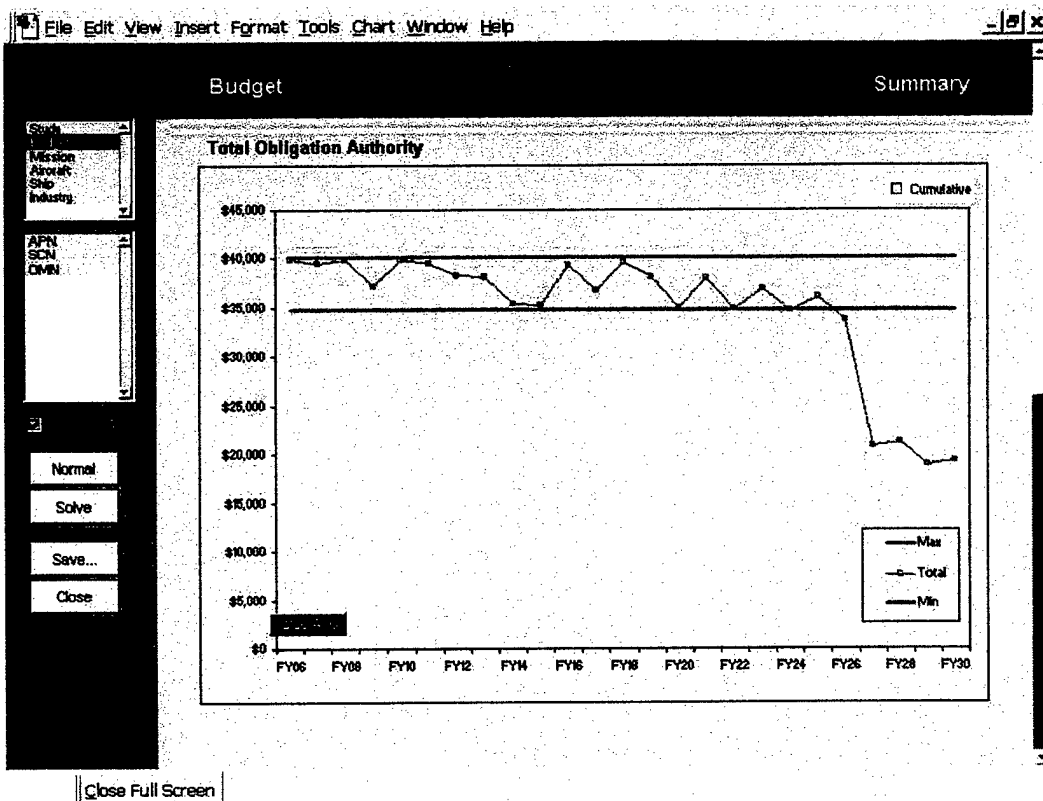
Figure 4.2: Budget summary (graphical view). The required budget to carry out the optimal plan is within the minimum and maximum levels except for the last years of the horizon due to end-effects.

Mission   Summary

Percent Achieved
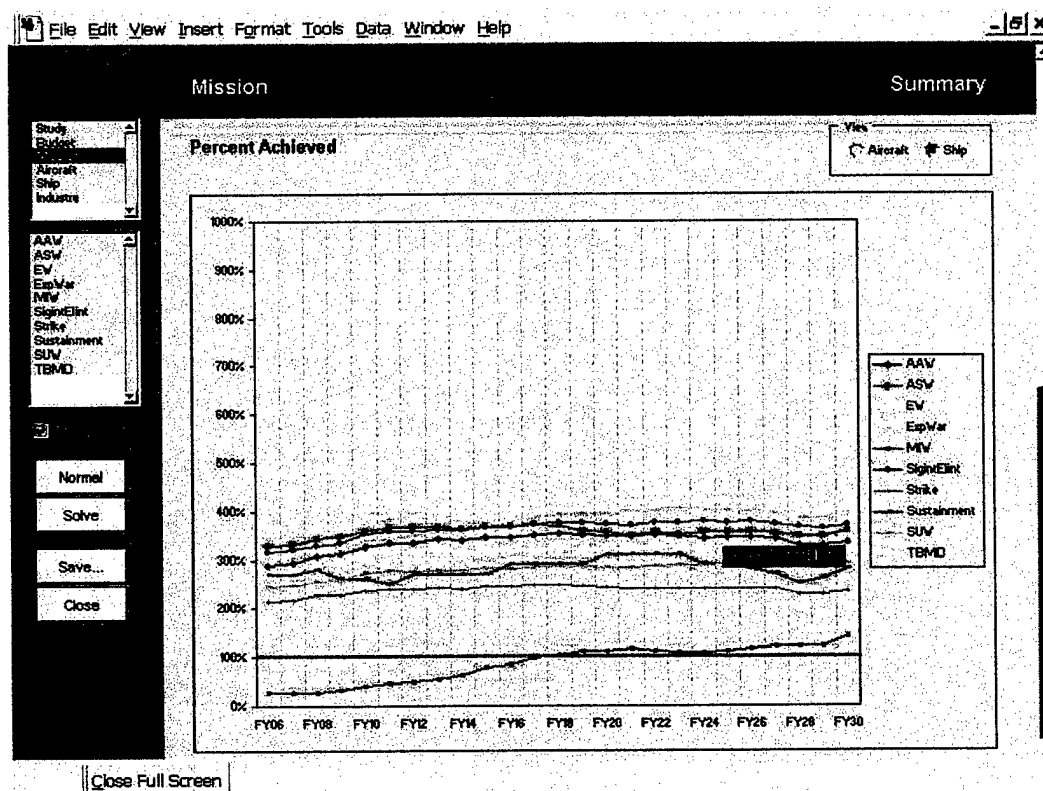
Figure 4.3: Mission summary screen for all requirements (graphical view). Available platforms exceed in most cases the required number of platforms.

Figure 4.4: SSN774 Inventory (data view). Procurement and retirement schedule.

Figure 4.5: SSN774 Production schedule at Eboat (data view). Yearly cost and labor required.
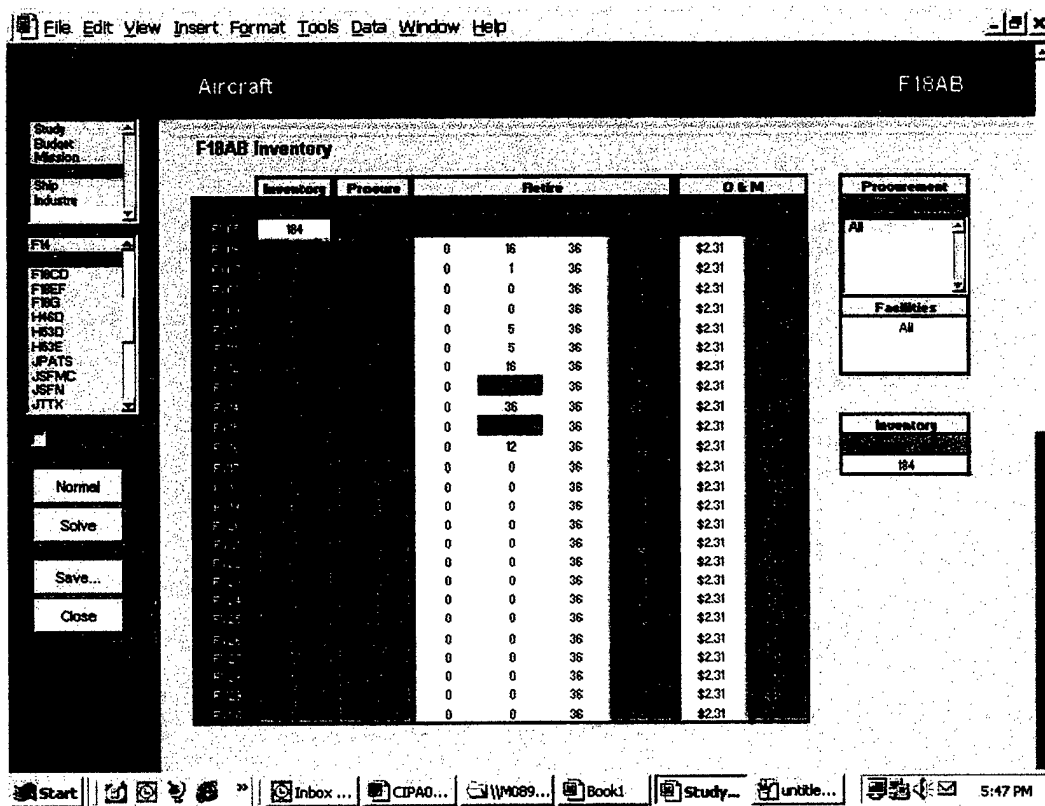
Figure 4.6: F18AB inventory (data view). Procurement and retirement schedule.
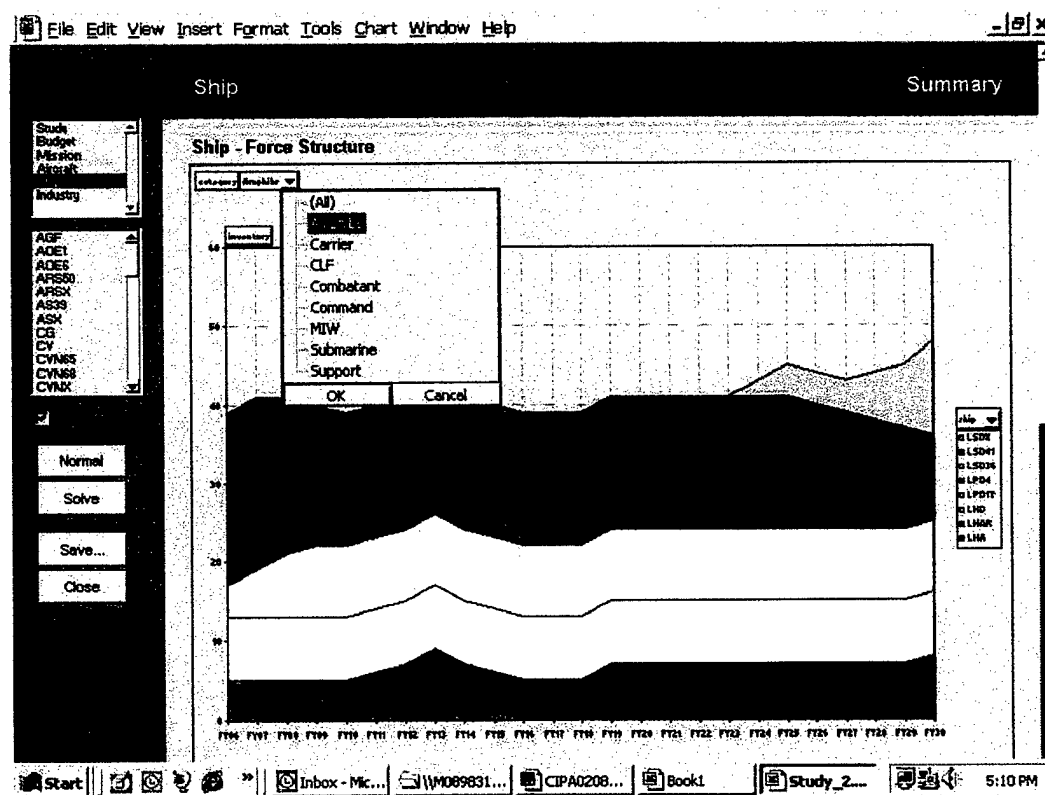
Figure 4.7: Amphibs inventory (graphical view).
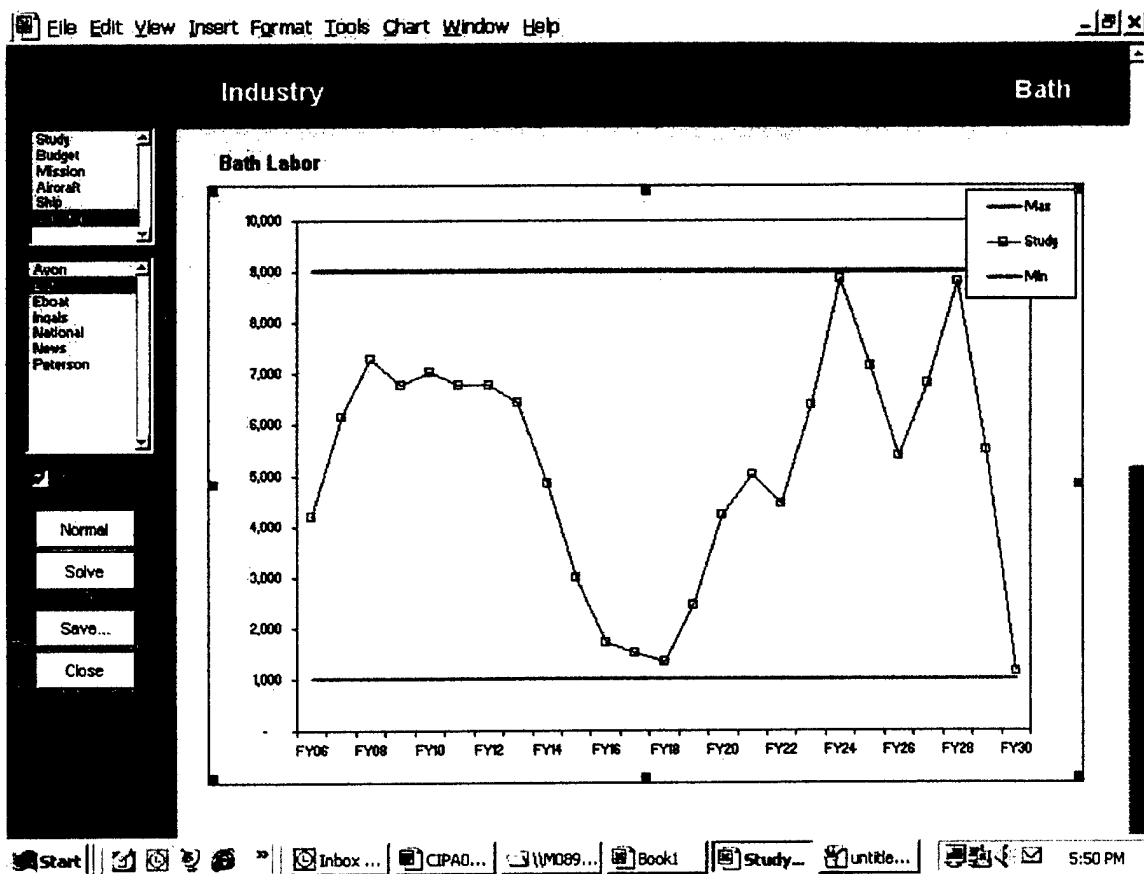
Figure 4.8: Labor at Bath shipyard (graphical view). Labor stays within the specified minimum and maximum number of workers.
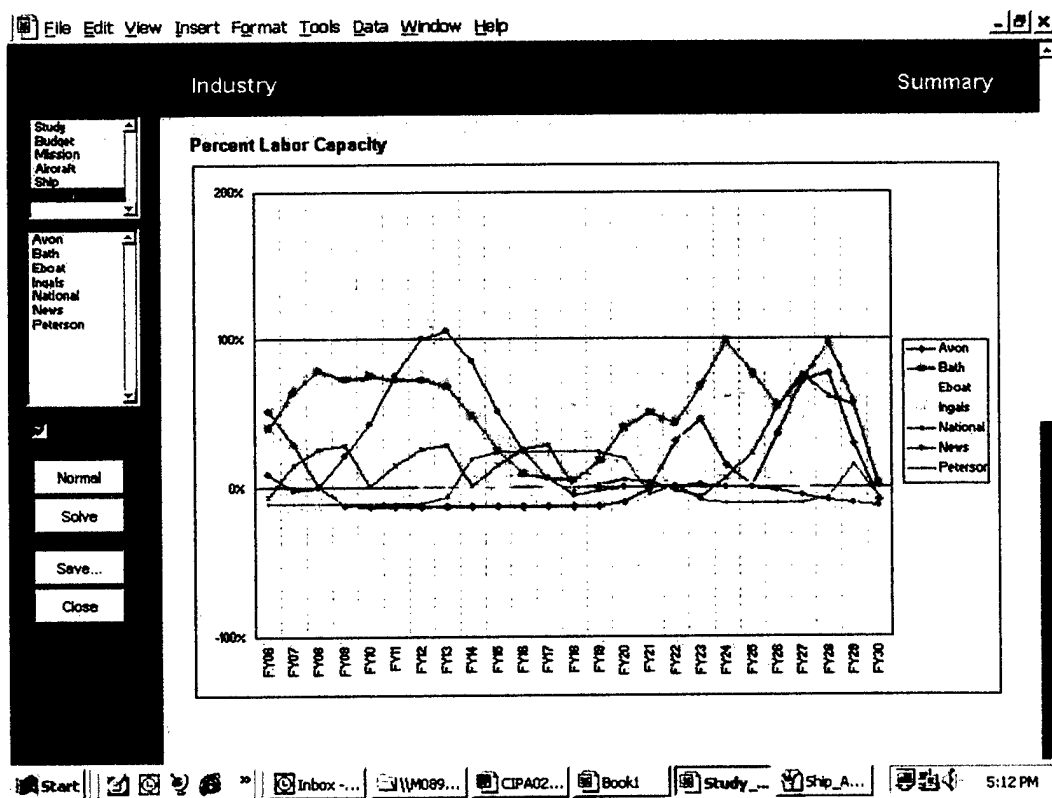
Figure 4.9: Labor for all shipyards (graphical view). For most of the years, shipyards stay within the specified labor limits.

Figure 4.10: Penalties. Unit cost for exceeding or falling under the limits in the specified category.

Figure 4.11: Detailed ship inventory report.

Figure 4.12: Detailed air inventory report.

Figure 4.13: Ship procurement plan report.

Figure 4.14: Aircraft procurement plan report.

Figure 4.15: SCN expenditure report.

Figure 4.16: APN expenditure report.

5. CIPA Solver(s)

5.1   Solvers

CIPA has two solvers: The heuristic solver (HS) described in Section 6 and the exact solver (ES) described in Section 7.

HS is a customized local-search heuristic that finds good solutions quickly. HS also provides a valid lower bound on the optimal solution cost—an objective assessment of the worst-case quality of the solution returned. Because it is very fast, the HS is always executed.

ES uses the commercial algebraic modeling language GAMS [Brooke et al., 1998] to generate a problem instance and then solves it with a contemporary commercial solver (e.g., OSL [GAMS/OSL, 2002], [OSL, 2002], CPLEX [GAMS/CPLEX, 2002], [ILOG, 2002], etc.). ES relaxes the planning problem by treating decisions for aircraft procurement and retirement and ship retirement as continuous, instead of discrete. Moreover, other requirements such as the squadron size for aircraft procurement are not considered. We post-process these solutions (see Section 7) and we provide results of some computation testing in Section 8.

The per-seat software license cost of ES is about $5,000. ES needs to be tended and used by an experienced modeler who can monitor and influence scenario run times, and detect failures. Accordingly, the role of ES is that of a high-cost calibration tool for the fast heuristic solver HS, and perhaps an option to be used selectively to thoroughly investigate and certify finalized scenarios before they are officially published.

5.2   Solver Framework

CIPA Solver integrates HS and (optionally) ES, communicating with the GUI by reading data, checking inconsistencies, and exporting results. See Figure 5.1.

41
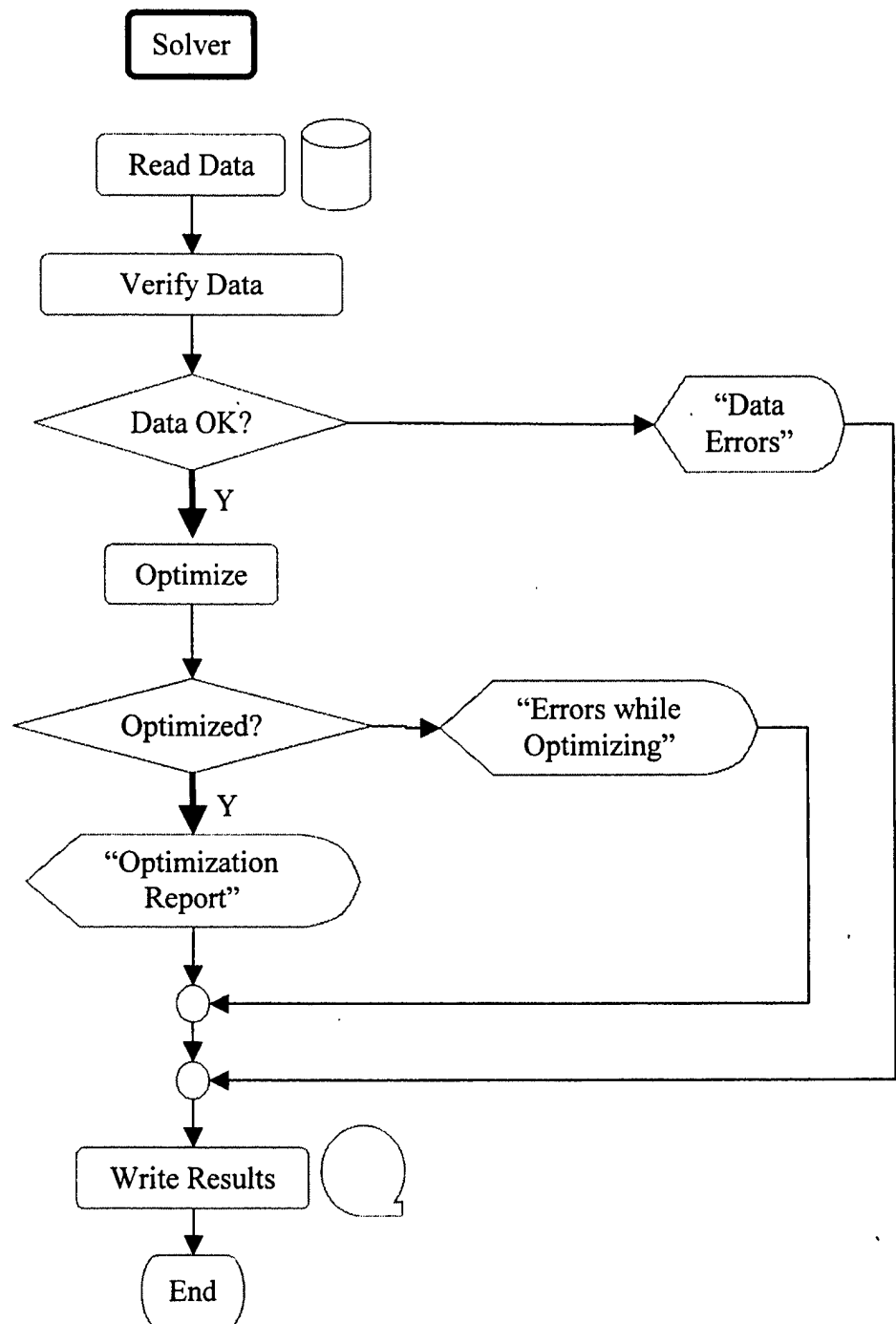
Figure 5.1: CIPA Solver Flowchart.

Figure 5.2 shows, in more detail, the steps involved in the Solve procedure:

1. First, we compute a lower bound (LB), which we call "Heuristic LB" that is based on individual bounds calculated for mission, labor and budget penalties. We may also (optionally) compute a lower bound by solving the linear programming relaxation of ES. We call this the "exact" lower bound, and it is typically better than the heuristic bound.

2. Next, we compute a heuristic upper bound (UB) (i.e., a feasible solution to the problem). This requires generating a (typically poor) initial solution that is enhanced during subsequent HS iterations of the so-called "Basic" and "Deep" search processes. These processes are described in detail in Section 6.

3. We can (optionally) proceed with ES with the expectation that results will take considerably longer to compute. To simplify the problem, we initially relax the integrality conditions for the aircraft procurement and retirement variables and for the ship retirement variables. We also disregard the condition to procure aircraft by squadrons. After the maximum (user controlled) allotted time, ES either returns an admissible solution, or not. A post-processing step rounds this solution to integer values for aircraft procurements and retirements and for the ship retirements. Then, a specialized procedure (see Section 7) adjusts aircraft procurement to meet the squadron size requirements. We resolve the CIPA model again and perform a final examination of the solution to guarantee it is feasible. This step is merely a security procedure, with minimum impact on computation time, since all the remaining control variables are determined by the main decision variables whose values we have just fixed.

4. Finally, the best lower bound and feasible solution are reported.

The Solver features a feasibility-checker and an objective function evaluator. The former is used to verify the feasibility of any candidate solution (that has not already been checked). This might include:

- any initial solution manually provided by the planner;
- the initial solution generated by HS;
- any successive candidate solution generated by HS; and/or
- the solution provided by ES.

Analogously, the objective function is evaluated for any candidate solution after its feasibility has been certified.

Figure 5.2: "Optimize" flowchart for the CIPA Solver. The steps involving the exact solver ES are optional.

6. CIPA Heuristic

In this section, we describe four HS modules: Initial Solution, Basic Search, Deep Search, and Lower Bound, using notation consistent with the CIPA formulation.

6.1   Initial Solution

The first HS step is to find an integer-feasible solution. Because CIPA constraints are endowed with elastic variables, with linear penalties for violations, it is always possible to assemble an integer-feasible solution, albeit with a lot of penalties.

The initial solution may be a direct user input or a solution found by HS. In the former case, the user's solution is checked for feasibility. If feasible, we compute its objective function value and proceed to Heuristic Basic Search. The rest of this section refers to the latter case when the user does not provide an initial solution or when that solution is infeasible.

We construct a myopic initial solution that assigns each variable the minimum value permitted by the constraints, according to the following scheme:

(1) Ship procurement: Produce at each shipyard the minimum amount of each ship class per year:

$$SPROC_{spyq} = \begin{cases} 1, & \text{if } q = \underline{sproc}_{spy} \\ 0, & \text{otherwise} \end{cases}, \forall s \in S, p \in P_s; \forall y \in Y$$

(2) Aircraft procurement: Procure the minimum feasible number allowed and meet the squadron size requirement:

- Find, for each $a \in A$ and $y \in Y$:

$$k_{ay} = \min\{k \in Z^+ \mid k \geq \underline{aproc}_{ay}, k = squad_a,$$

$$\exists \tilde{i}_{ay} \in I_a \text{ such that } \underline{inc}_{ay\tilde{i}_{ay}} \leq k \leq \overline{inc}_{ay\tilde{i}_{ay}} \}$$

where $k = squad_a$ denotes the condition "k is a multiple of $squad_a$"
(Note that typically k_{ay} will be zero unless $\underline{aproc}_{ay} > 0$ or $\underline{inc}_{ay1} > 0$)

- Assign: $APROC_{ayi} = \begin{cases} k_{ay}, & \text{if } i = \tilde{i}_{ay} \\ 0, & \text{otherwise} \end{cases} \forall a \in A, \forall y \in Y$

(3) Ship retirement: Retire the minimum of individual and cumulative requirements. Because cumulative minima in future years may require larger retirements in previous years, we need to first compute the "actual" cumulative minimum retirement implicit in the initial data:

45

- Starting with $Actual\_csret_{s1} := \max\{csret_{s1}, \underline{sret}_{s1}\}$, compute

$$Actual\_csret_{sy} := \max\{csret_{sy}, Actual\_csret_{s,y-1} + \underline{sret}_{sy}\}$$

- Starting at $y = |Y| - 1$ and working backwards, update:

$$Actual\_csret_{sy} := \max\{Actual\_csret_{sy}, Actual\_csret_{s,y+1} - \overline{sret}_{s,y+1}\}$$

- Starting with $SRET_{s1} = Actual\_csret_{s1}$, compute the definite

$$SRET_{sy} := Actual\_csret_{sy} - Actual\_csret_{s,y-1}$$

(4) Aircraft retirement: Retire the minimum of individual and cumulative requirements. Because the cumulative minima in future years may require larger retirements in previous years, we need to first compute the "actual" cumulative minimum retirement implicit in the initial data:

- Starting with $Actual\_caret_{a1} := \max\{caret_{a1}, \underline{aret}_{a1}\}$, compute

$$Actual\_caret_{ay} := \max\{caret_{ay}, Actual\_caret_{a,y-1} + \underline{aret}_{ay}\}$$

- Starting at $y = |Y| - 1$ and working backwards, update:

$$Actual\_caret_{ay} := \max\{Actual\_caret_{ay}, Actual\_caret_{a,y+1} - \overline{aret}_{a,y+1}\}$$

- Starting with $ARET_{a1} = Actual\_caret_{a1}$, compute the definite

$$ARET_{ay} := Actual\_caret_{ay} - Actual\_caret_{a,y-1}$$

6.2 Basic Search

The CIPA objective function has three main penalty categories: mission shortfall, budget deficit and surplus, and work-force (industrial) shortfall and excess.

Figure 6.1 displays how CIPA decisions impact the CIPA objective function value. For example, changing the number of ships delivered from a shipyard in a given year changes:

- the labor used at the shipyard several years prior to delivery (thus potentially changing the industrial penalty);

- the number of ships available to perform mission(s) from the year of delivery onwards (and thus potentially changing the mission penalty); and

- the SCN and O&M costs (thus potentially changing the budget penalty).

46
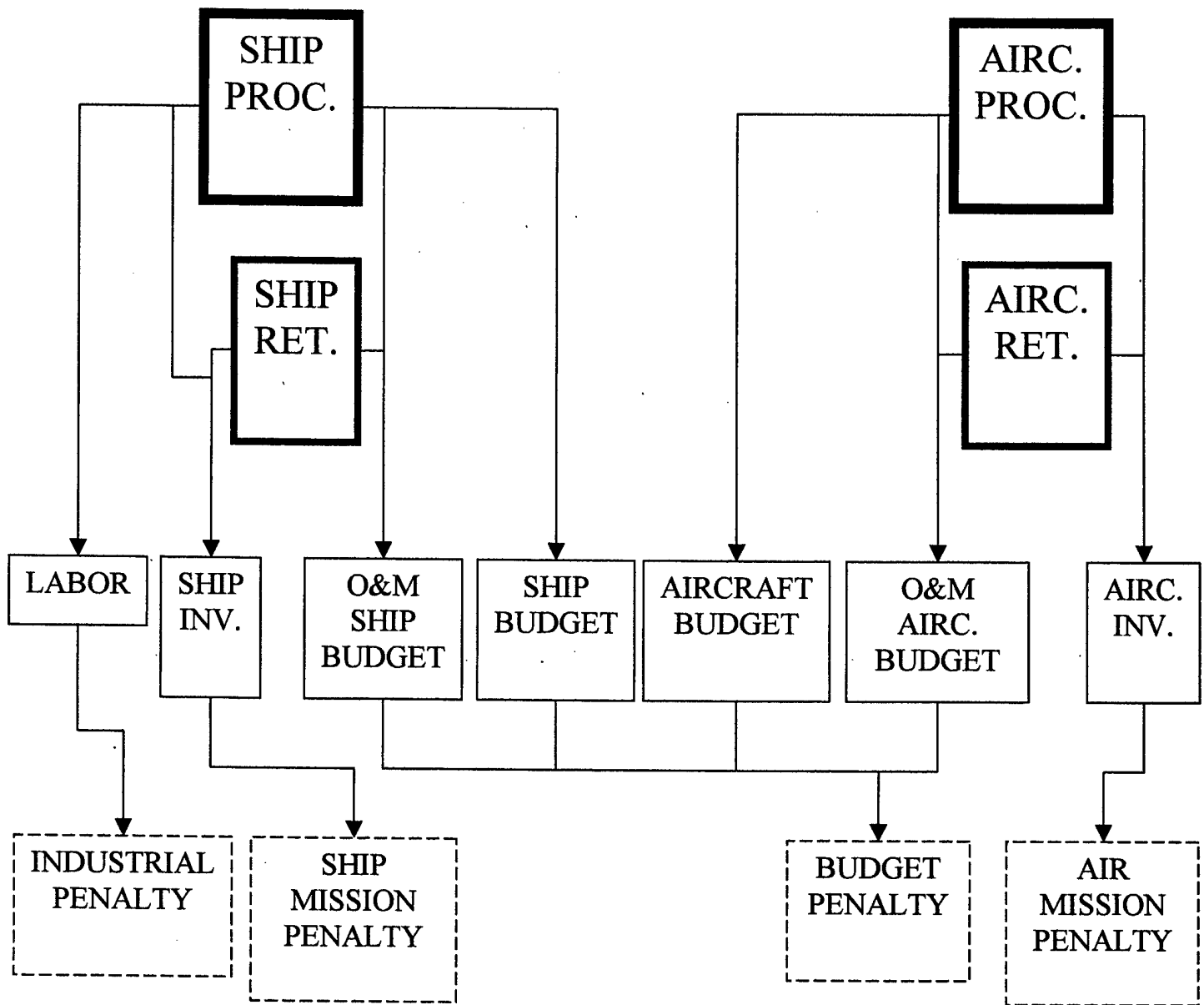
Figure 6.1: Influence implications of the CIPA objective function. The CIPA objective function has three main penalty categories (Ship-Mission and Air-Mission penalties have been separated for the clarity of exposition), and each decision potentially influences a subset of these.

We restrict our analysis to the search for the best possible configuration of a vector $\mathbf{x}$, where:

$\mathbf{x}$=(ship procurement, aircraft procurement, ship retirement, aircraft retirement).

We consider our objective function F divided into three components: mission penalty, budget penalty, and labor penalty. That is, for any feasible solution $\mathbf{x}$:

$$F(\mathbf{x}) = F^M(\mathbf{x}) + F^B(\mathbf{x}) + F^L(\mathbf{x})$$

where F^M, F^B and F^L are the three aforementioned components, respectively. Moreover, the first two terms of the sum can be decomposed as follows:

$$F^M(\mathbf{x}) = F^{SM}(\mathbf{x}) + F^{AM}(\mathbf{x}) \text{ and}$$
$$F^B(\mathbf{x}) = F^{YB}(\mathbf{x}) + F^{CB}(\mathbf{x})$$

where $F^{SM}(\mathbf{x})$ and $F^{AM}(\mathbf{x})$ are the ship-mission and air-mission penalties, and $F^{YB}(\mathbf{x})$ and $F^{CB}(\mathbf{x})$ are the yearly budget and cumulative-budget penalties, respectively.

Our local search seeks feasible solutions that progressively improve the objective function value. We accomplish this by evaluating multiple synchronous modifications to the incumbent configuration, and selecting those that lead to better solutions.

Figure 6.2 shows the basics of the local search procedure. We start with an initial incumbent solution $\mathbf{x}^*$. Then, we apply a number of "Search Strategies" indexed by $ss=1,...,SS$. Each strategy is characterized by generating a number of tentative solutions, $\mathbf{x}^m$, $m=1,...,M$, located in the "neighborhood" of the current incumbent $\mathbf{x}^*$.

Each $\mathbf{x}^m$ is assigned a ranking position consistent with the strategy, $R_{ss}(\mathbf{x}^m)$, where a negative ranking means that the solution is deemed worse that the current $\mathbf{x}^*$ (whose ranking is zero). Among these candidates, we select the one (say $\mathbf{x}^k$) yielding the best ranking. If this improves the current incumbent, we update $\mathbf{x}^*$ and continue to generate new tentative solutions using the same strategy. If not, we switch to a new strategy to generate the next group of candidates, repeating the process until no strategy produces an improvement.

A straightforward ranking assignment is given by $R_{ss}(\mathbf{x}^m)=F(\mathbf{x}^*)-F(\mathbf{x}^m)$. However, we will employ other ranking schemes (described later in this document) depending on the strategy ss and on each specific component of the objective function, $F^{SM}(\mathbf{x})$, $F^{AM}(\mathbf{x})$, $F^{YB}(\mathbf{x})$, $F^{CB}(\mathbf{x})$ and $F^L(\mathbf{x})$.

The success of Basic Search depends on the strategies applied. We present four strategies that, in practice, have shown a remarkable ability to solve our CIPA model: We call these "Mission," "Labor," "Budget," and "Retirement."

48
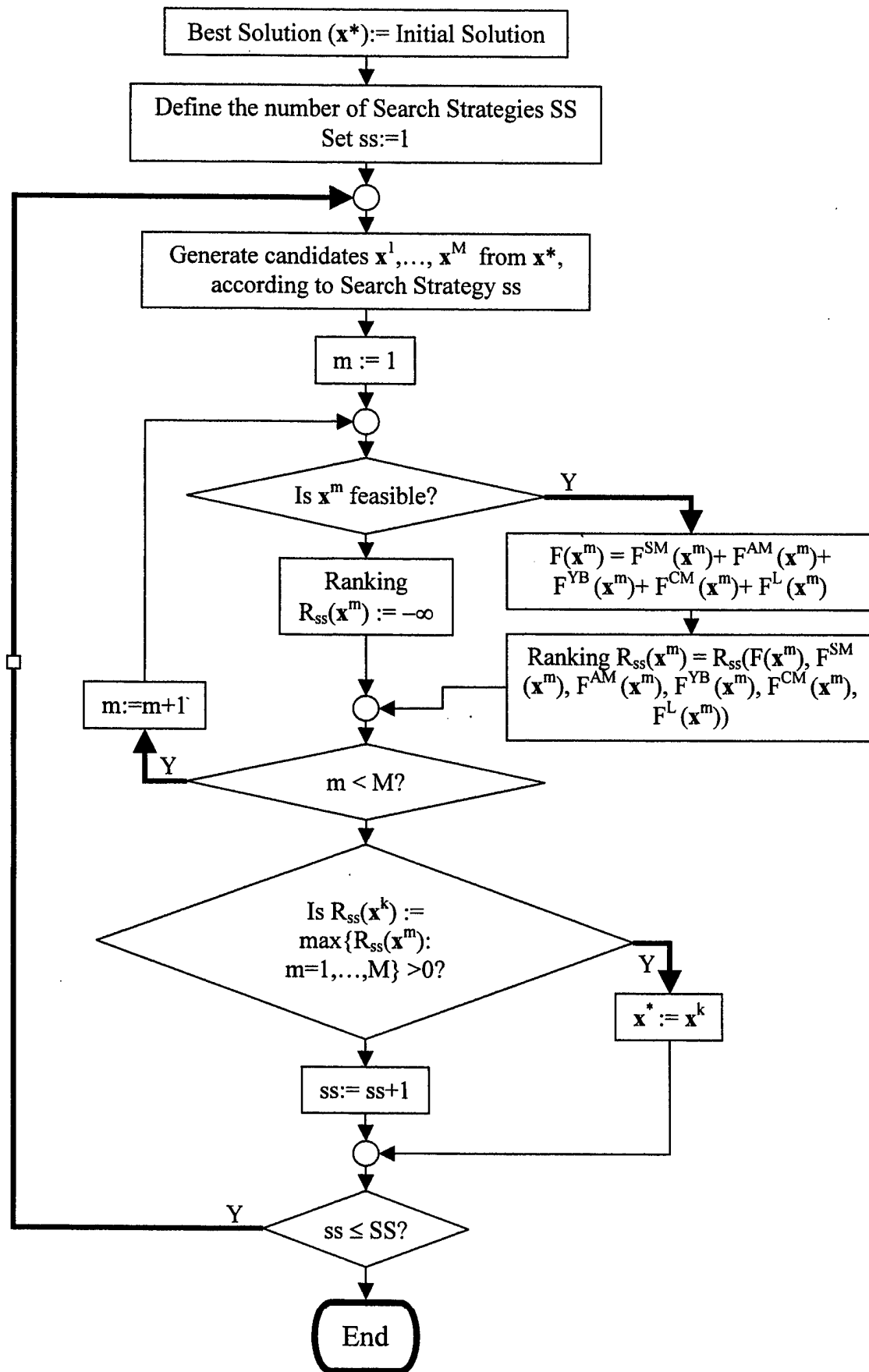
Figure 6.2: Flowchart of Basic Search.

49

6.2.1 Mission Search Strategy

Mission requirements drive the CIPA procurement and retirement decisions. For this reason, our first search consists of reducing the penalties caused by mission shortfalls, that is, the $F^M(x) = F^{SM}(x) + F^{AM}(x)$ portion of the total penalty $F(x)$. The mission strategy encompasses two sub-strategies: "Ship-Mission" and "Air-Mission." As depicted in Figures 6.3.a-b, we incorporate new ship procurements of a given class "s" and new aircraft procurements of type "a" at a time, throughout the years, without making any other change to the incumbent solution.
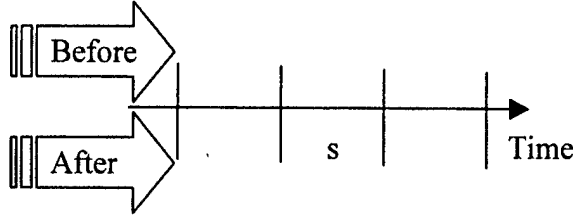
Figure 6.3.a: Ship-Mission strategy.
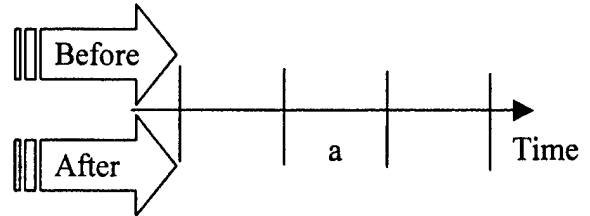
Figure 6.3.b: Air-Mission strategy.

Formally, the definition of this strategy is as follows:

1. Find the larger of the two weighted penalties $w^{SM} F^{SM}(x)$ and $w^{AM} F^{AM}(x)$, where w^{SM} and w^{AM} are given weights (see "Remarks" on the next page for details). If $w^{SM} F^{SM}(x) > w^{AM} F^{AM}(x)$, use the "Ship-Mission" sub-strategy, otherwise, use "Air-Mission."

2. If the selected sub-strategy is Ship-Mission, define the following new candidates, x^m, and their rankings, $R_{SM}(x^m)$, for $m=1,...,M$:

   - Select $m = (\hat{s}, \hat{p}, \hat{y}, \hat{q})$ where $\hat{s} \in S$, $\hat{p} \in P_{\hat{s}}$, $\hat{y} \in Y$, $\hat{q} \in Q_{\hat{s}\hat{p}\hat{y}}$, $\hat{q} \geq 1$.
   - Assign the following components of x^m:

     (a) Ship retirement, Aircraft procurement and retirement: Same as in x^*

     (b) Ship procurement: $SPROC_{spyq} := \begin{cases} 1, & \text{if } s = \hat{s}, p = \hat{p}, y = \hat{y}, q = \hat{q} \\ 0, & \text{if } s = \hat{s}, p = \hat{p}, y = \hat{y}, q \neq \hat{q} \\ \text{same as in } x^*, & \text{otherwise} \end{cases}$

   - Assign the Ranking function as follows:

$$R_{SM}(x^m) = \begin{cases} 0, & \text{if } F^{SM}(x^*) < F^{SM}(x^m) \text{ or } F(x^*) < F(x^m) \\ w_{SM}^{SM}\left(F^{SM}(x^*) - F^{SM}(x^m)\right) + w_{SM}^L\left(F^L(x^*) - F^L(x^m)\right)^+ + \\ \quad w_{SM}^B\left(F^{YB}(x^*) - F^{YB}(x^m) + F^{CB}(x^*) - F^{CB}(x^m)\right)^+, & \text{otherwise} \end{cases}$$

50

where all the $w_{SM}^{(\cdot)}$ are weights to assign different leverage to each change in the penalties. Also, $(.)^+$ refers to the positive part of the argument, that is, the argument (if positive) or zero (otherwise).

3. If the sub-strategy is Air-Mission, define the following new candidates, x^m, and their rankings, $R_{AM}(x^m)$, for $m=1,...,M$:

- Select $m = \left(\hat{a}, \hat{y}, \hat{i}, \hat{k}\right)$ where $\hat{a} \in A$, $\hat{y} \in Y$, $\hat{i} \in I_{\hat{a}}$, $\underline{inc}_{\hat{a}\hat{y}\hat{i}} \leq \hat{k} \leq \overline{inc}_{\hat{a}\hat{y}\hat{i}}$ and $0 \neq \hat{k} = squad_{\hat{a}}$

- Assign the following components of x^m:

  (a) Ship procurement and retirement, Aircraft retirement: Same as in $\mathbf{x^*}$.

  (b) Aircraft procurement: $APROC_{ayi} := \begin{cases} \hat{k}, & \text{if } a = \hat{a}, y = \hat{y}, i = \hat{i} \\ 0, & \text{if } a = \hat{a}, y = \hat{y}, i \neq \hat{i} \\ \text{same as in } x^*, & \text{otherwise} \end{cases}$

- Assign the Ranking function as follows:

$$R_{AM}\left(x^m\right) = \begin{cases} 0, \text{ if } F^{AM}\left(x^*\right) < F^{AM}\left(x^m\right) \text{ or } F\left(x^*\right) < F\left(x^m\right) \\ w_{AM}^{AM}\left(F^{AM}\left(x^*\right) - F^{AM}\left(x^m\right)\right) + \\ \quad w_{AM}^{B}\left(F^{YB}\left(x^*\right) - F^{YB}\left(x^m\right) + F^{CB}\left(x^*\right) - F^{CB}\left(x^m\right)\right)^+, \text{ otherwise} \end{cases}$$

where all the $w_{AM}^{(\cdot)}$ are weights to assign different leverage to each change in the penalties.

Remark 1: In the Ship-Mission sub-strategy, we allow changes that increase the labor and budget penalties if, in return, both the Ship-Mission penalty and the total penalty are reduced. In the Air-Mission sub-strategy, we allow changes that increase the budget penalty if, in return, both the Air-Mission penalty and the total penalty are reduced.

Remark 2: If $w^{SM} F^{SM}(x) > w^{AM} F^{AM}(x)$ but the Ship-Mission sub-strategy yields no improvement, we proceed to the Air-Mission sub-strategy, and vice versa. The Mission strategy terminates when neither sub-strategy improves the current solution.

Remark 3: In practice, the whole Mission strategy is executed twice. The first time, we consider a fictitious big penalty for budget excess, to encourage the purchase of platforms without exceeding the maximum budget. In the second run, the actual penalties (α_y^{B+} and α_y^{CB+} in the model) are used, seeking a solution that may benefit from budget flexibility.

Remark 4: Our typical settings for the weights used in this strategy are as follows:

$$w^{SM} = 1.0, w^{AM} = 5.0 \; ; \; w_{SM}^{SM} = 10.0, w_{SM}^{L} = 3.0, w_{SM}^{B} = 1.0 \; ; \; w_{AM}^{AM} = 10.0, w_{AM}^{B} = 1.0$$

These weights are only for the purpose of the algorithm. Violations in the objective function are measured in $/unaccomplished mission (for mission violations), $/worker (for labor limit violations) and $/$ (for budget violations). Our weights here ($/$) represent how much a violation in any of these categories is offset by a benefit in a different category. In particular, we give more weight to ship-missions and air-missions because the incumbent strategy aims to decrease the total mission penalty. Thus, we favor a decision that reduces our mission penalty by $10 and the labor penalty by $30 (the total score of this decision is $10 \times \$10 + 3 \times \$20 = \$160$) rather than a decision that reduces the mission penalty by $5 and the labor by $30 ($10 \times \$5 + 3 \times \$30 = \$140$). These weights are modified for other strategies to accommodate their own goals.

6.2.2 Labor Search Strategy

At this point, even though no additional purchase of ships is recommended by the Ship-Mission search strategy, we may still need to increase ship production in order to reduce industry penalties. This situation occurs when the plants are under-employed and the remaining budget permits procuring more platforms. As in the Ship-part of the Mission strategy, we only consider ship procurements of a given class "s" at a time, throughout the years (Figure 6.4).
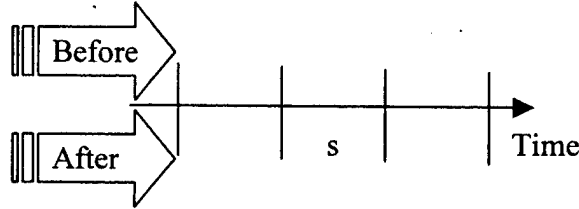
Figure 6.4: Labor strategy.

The Labor strategy definition is as follows:

Define the following new candidates, x^m, and their rankings, $R_L(x^m)$, for $m=1,...,M$:

- Select $m = (\hat{s}, \hat{p}, \hat{y}, \hat{q})$ where $\hat{s} \in S$, $\hat{p} \in P_{\hat{s}}$, $\hat{y} \in Y$, $\hat{q} \in Q_{\hat{s}\hat{p}\hat{y}}$, $\hat{q} \geq 1$.
- Assign the following components of x^m:

> (a) Ship retirement, Aircraft procurement and retirement: Same as in x^*.
>
> (b) Ship procurement: $SPROC_{spyq} := \begin{cases} 1, & \text{if } s = \hat{s}, p = \hat{p}, y = \hat{y}, q = \hat{q} \\ 0, & \text{if } s = \hat{s}, p = \hat{p}, y = \hat{y}, q \neq \hat{q} \\ \text{same as in } x^*, & \text{otherwise} \end{cases}$

- Assign the Ranking function as follows:

$$R_L\left(x^m\right)=\begin{cases}0, \text{ if } F^L\left(x*\right)<F^L\left(x^m\right) \text{ or } F\left(x*\right)<F\left(x^m\right)\\ w_L^{SM}\left(F^{SM}\left(x*\right)-F^{SM}\left(x^m\right)\right)^+ \;+\; w_L^L\left(F^L\left(x*\right)-F^L\left(x^m\right)\right) \;+\\ \quad w_L^B\left(F^{YB}\left(x*\right)-F^{YB}\left(x^m\right)+F^{CB}\left(x*\right)-F^{CB}\left(x^m\right)\right)^+, \text{ otherwise}\end{cases}$$

where all the $w_L^{(\cdot)}$ are weights to assign different leverage to each change in the penalties.

Remark 1: In the Labor strategy, we allow changes that increase the ship-mission and budget penalties if, in return, both the labor penalty and the total penalty are reduced.

Remark 2: Our typical settings for the weights used in this strategy are as follows:

$$w_L^{SM}=5.0, w_L^L=10.0, w_L^B=1.0$$

6.2.3 Budget Search Strategy

In some years, our expenditures may be under the minimum limit, even after having settled mission and labor requirements. In this case, to avoid incurring budget penalties, we may acquire extra platforms to increase these expenditures. Because our only purpose here is to spend the spare money, we check first with ship procurements and then with aircraft procurements. As in the previous strategies, we consider ship and aircraft procurement throughout the years (Figures 6.5.a-b).
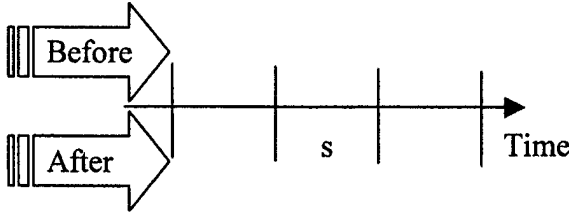
Figure 6.5.a: Budget-Ship strategy.   **Figure 6.5.b:** Budget-Aircraft strategy.
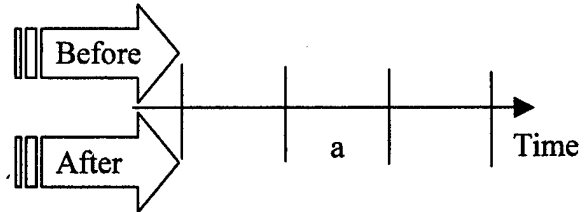
The Budget Strategy is defined as follows:

1. Budget (Ship procurement part): Define the following new candidates, x^m, and their rankings, $R_B(x^m)$, for $m=1,...,M$:

   - Select $m=\left(\hat{s},\hat{p},\hat{y},\hat{q}\right)$ where $\hat{s}\in S$, $\hat{p}\in P_{\hat{s}}$, $\hat{y}\in Y$, $\hat{q}\in Q_{\hat{s}\hat{p}\hat{y}}$, $\hat{q}\geq 1$.
   - Assign the following components of x^m:

      (a) Ship retirement, Aircraft procurement and retirement: Same as in **x***.

(b) Ship procurement: $SPROC_{spyq} := \begin{cases} 1, & \text{if } s = \hat{s}, p = \hat{p}, y = \hat{y}, q = \hat{q} \\ 0, & \text{if } s = \hat{s}, p = \hat{p}, y = \hat{y}, q \neq \hat{q} \\ \text{same as in } x^*, & \text{otherwise} \end{cases}$

- Assign the Ranking function as follows:

$$R_B\left(x^m\right) = \begin{cases} 0, & \text{if } F^{YB}(x^*) + F^{CB}(x^*) < F^{YB}\left(x^m\right) + F^{CB}\left(x^m\right) \text{ or } F(x^*) < F\left(x^m\right) \\ F^{YB}(x^*) - F^{YB}\left(x^m\right) + F^{CB}(x^*) - F^{CB}\left(x^m\right), & \text{otherwise} \end{cases}$$

2. **Budget (Aircraft procurement part):** Define the following new candidates, x^m, and their rankings, $R_B(x^m)$, for $m=1,...,M$:

   - Select $\cdot m = \left(\hat{a}, \hat{y}, \hat{i}, \hat{k}\right)$ where $\hat{a} \in A$, $\hat{y} \in Y$, $\hat{i} \in I_{\hat{a}}$, $\underline{inc}_{\hat{a}\hat{y}\hat{i}} \leq \hat{k} \leq \overline{inc}_{\hat{a}\hat{y}\hat{i}}$ and $0 \neq \hat{k} = squad_{\hat{a}}$.

   - Assign the following components of x^m:

     (a) Ship procurement and retirement, Aircraft retirement: Same as in x^*.

     (b) Aircraft procurement: $APROC_{ayi} := \begin{cases} \hat{k}, & \text{if } a = \hat{a}, y = \hat{y}, i = \hat{i} \\ 0, & \text{if } a = \hat{a}, y = \hat{y}, i \neq \hat{i} \\ \text{same as in } x^*, & \text{otherwise} \end{cases}$

   - Assign the ranking function as follows:

$$R_B\left(x^m\right) = \begin{cases} 0, & \text{if } F^{YB}(x^*) + F^{CB}(x^*) < F^{YB}\left(x^m\right) + F^{CB}\left(x^m\right) \text{ or } F(x^*) < F\left(x^m\right) \\ F^{YB}(x^*) - F^{YB}\left(x^m\right) + F^{CB}(x^*) - F^{CB}\left(x^m\right), & \text{otherwise} \end{cases}$$

Remark: The ranking function R_B is the same for the ship and aircraft stages of the budget search strategy. We only allow changes that reduce both the budget penalty and the total penalty.

6.2.4 Retirement Search Strategy

The initial retirement schedule follows minimum retirements from the input data (which may be zero), disregarding how new procurements should influence earlier retirements of obsolete or redundant assets. The Retirement Strategy assesses the tradeoff between earlier platform retirements (which reduces O&M costs) and Ship-Mission accomplishment. The analysis is done for ships first, and then for aircraft (Figures 6.6.a-b).

54
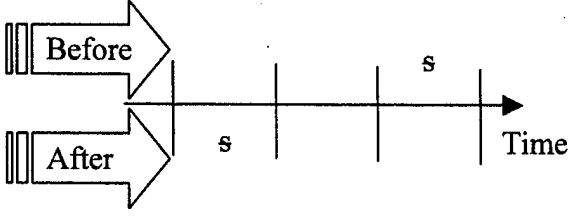
Figure 6.6.a: Ship-Retirement strategy. **Figure 6.6.b:** Air-Retirement strategy.
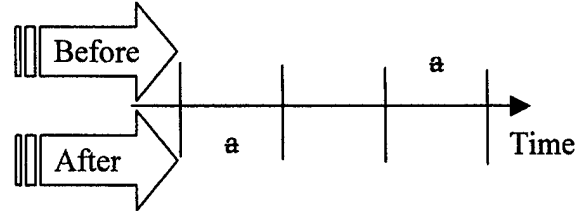
This strategy definition is:

1. Retirement (Ship part): Define the following new candidates, x^m, and their rankings, $R_{RS}(x^m)$, for $m=1,...,M$:

- Select $m = \left(\hat{s}, \hat{y}, \hat{y}'\right)$ where $\hat{s} \in S$, $\hat{y} \in Y$, $SRET_{\hat{s},\hat{y}} \geq 1, \hat{y}' \in Y$, $\hat{y}' \leq \hat{y} - 1$
- Assign the following components of x^m:

    (a) Ship procurement and Aircraft procurement and retirement: Same as in $\mathbf{x}^*$

$$(b) \quad SRET_{sy} := \begin{cases} SRET_{sy} - 1, & \text{if } s = \hat{s}, y = \hat{y} \\ SRET_{sy} + 1, & \text{if } s = \hat{s}, y = \hat{y}' \\ \text{same as in } x^*, & \text{otherwise} \end{cases}$$

Remark: Notice that $SRET_{sy}$ appears on both sides of the expression above. The one on the left-hand side refers to the new value to be assigned to $SRET_{sy}$. This depends on the former value, $SRET_{sy}$, that appears on the right-hand side. Hereafter, we assume this notation.

- Assign the Ranking function as follows:

$$R_{RS}\left(x^m\right) = \begin{cases} 0, & \text{if } F\left(x^*\right) < F\left(x^m\right) \\ w_{RS}^{SM}\left(F^{SM}\left(x^*\right) - F^{SM}\left(x^m\right)\right) + w_{RS}^{OM}\left(SOMC(x^*) - SOMC\left(x^m\right)\right), & \text{otherwise} \end{cases}$$

where $SOMC(.)$ is the ship operation and maintenance cost for the solution in the argument, and all the $w_{RS}^{(.)}$ are weights to assign different leverage to each component of the ranking.

2. Retirement (Aircraft part): Define the following new candidates, x^m, and their rankings, $R_{RA}(x^m)$, for $m=1,...,M$:

- Select $m = \left(\hat{a}, \hat{y}, \hat{y}'\right)$ where $\hat{a} \in A$, $\hat{y} \in Y$, $ARET_{\hat{s},\hat{y}} \geq squad_{\hat{a}}, \hat{y}' \in Y$, $\hat{y}' \leq \hat{y} - 1$.

- Assign the following components of x^m:

(a) Ship procurement, Aircraft procurement and retirement: Same as in x^*.

(b) Ship retirement: $ARET_{ay} := \begin{cases} ARET_{ay} - squad_a, & \text{if } a = \hat{a}, y = \hat{y} \\ ARET_{ay} + squad_a, & \text{if } a = \hat{a}, y = \hat{y}' \\ \text{same as in } x^*, & \text{otherwise} \end{cases}$

- Assign the Ranking function as follows:

$$R_{RA}\left(x^m\right) = \begin{cases} 0, & \text{if } F(x^*) < F\left(x^m\right) \\ w_{RA}^{AM}\left(F^{AM}(x^*) - F^{AM}\left(x^m\right)\right) + w_R^{OM}\left(AOMC(x^*) - AOMC\left(x^m\right)\right), & \text{otherwise} \end{cases}$$

where $AOMC(.)$ is the aircraft operation and maintenance cost for the solution in the argument, and all the $w_{RA}^{(.)}$ are weights to assign different leverage to each component of the ranking.

Remark 1: We allow changes that may increase the ship and air mission penalties if, in return, the total penalty is reduced.

Remark 2: In practice, we use this strategy before and after each of the other strategies (Mission, Labor, and Budget). This permits retirements to keep pace as the Basic Search updates procurements.

Remark 3: Our typical settings for the weights used in this strategy are:

$$w_{RS}^{SM} = 1.0, w_{RS}^{OM} = 1.0 ; \quad w_{RA}^{AM} = 1.0, w_{RA}^{OM} = 1.0$$

6.3 Deep Search

In Basic Search, we explore new solutions that differ from the best incumbent solution in only one component. In Retirement Search, each change affects two components, with a retirement moved earlier. Note, however, that each change involves just one platform type at a time.

The Basic Search strategy is preserved, in part, during Deep Search: We continue to list a number of candidate moves and select the one with the best ranking. However, Deep Search provides a broader spectrum of configurations to analyze in hopes of overcoming the myopia of Basic Search. Theoretically, Deep Search can implement any conceivable move, whether it consists of a single change in the components or combines multiple changes.

56

Of course, by increasing the number of "neighbors" to explore we increase runtime. For this reason, we have implemented a limited number of strategies in Deep Search, namely "Joint Retirement-Procurement" and "Platform Exchange," which we present in the remainder of this section.

6.3.1 Joint Retirement-Procurement Deep Search Strategy

Basic Search may skip some beneficial moves such as the purchase of new platforms in exchange for ageing ones. If their mission capabilities are similar, this type of exchange may be worthwhile because of the savings in O&M costs. This subtlety may be overlooked by Basic Search if, during the Mission search strategy, there is no deficiency in mission coverage, and therefore no need to procure a new platform. During the Retirement search strategy, it is not advisable to retire a platform that is carrying out a mission. Because the two search strategies do not cooperate, this move would evade Basic Search.

To overcome this difficulty, we define a Joint Retirement-Procurement Deep Search Strategy that incorporates a slight modification of the idea above (Figures 6.7.a-b): We compare alternatives for advancing platform retirements, while compensating for this by advancing new platforms of similar characteristics:
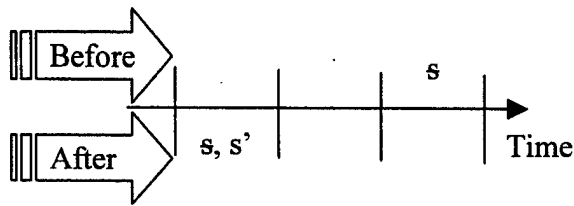
Figure 6.7.a: Joint Ret.-Proc. (Ship) strategy. **Figure 6.7.b:** Joint Ret.-Proc. (Aircraft) strategy.
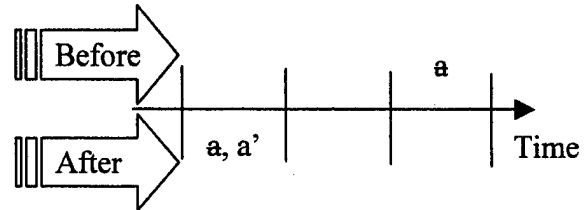
1. Joint Procurement-Retirement (Ship part): Define the following new candidates, x^m, and their rankings, $R_{PR}(x^m)$, for $m=1,\dots,M$:

   - Select $m = (\hat{s}, \hat{s}', \hat{p}', \hat{y}, \hat{y}')$ where $\hat{s}, \hat{s}' \in S$, $\hat{p}' \in P_{\hat{s}'}$,
     $\hat{y}, \hat{y}' \in Y, SRET_{\hat{s}\hat{y}} \geq 1$, $\hat{y}' \leq \hat{y} - 1$.

   - Assign the following components of x^m:

     (a) Aircraft procurement and retirement: Same as in **x***.

     (b) Ship retirement: $SRET_{sy} := \begin{cases} SRET_{sy} - 1, & \text{if } s = \hat{s}, y = \hat{y} \\ SRET_{sy} + 1, & \text{if } s = \hat{s}, y = \hat{y}' \\ \text{same as in } x^*, & \text{otherwise} \end{cases}$

57

(c) Ship procurement:

$$SPROC_{spyq} := \begin{cases} 1, & \text{if } s = \hat{s}', p = \hat{p}', y = \hat{y}', SPROC_{s'p'y'q-1} = 1 \\ 0, & \text{if } s = \hat{s}', p = \hat{p}', y = \hat{y}', SPROC_{s'p'y'q} = 1 \\ \text{same as in } x^*, & \text{otherwise} \end{cases}$$

- Assign the Ranking function as follows:

$$R_{JR}(x^m) = \begin{cases} 0, & \text{if } F(x^*) < F(x^m) \\ F(x^*) - F(x^m), & \text{otherwise} \end{cases}$$

2. Joint Procurement-Retirement (Aircraft part): Define the following new candidates, x^m, and their rankings, $R_{PR}(x^m)$, for $m=1,...,M$:

- Select $m = (\hat{a}, \hat{a}', \hat{y}, \hat{y}')$ where $\hat{a}, \hat{a}' \in A$, $\hat{y}, \hat{y}' \in Y$, $SRET_{\hat{s}\hat{y}} \geq squad_{\hat{a}}$, $\hat{y}' \leq \hat{y} - 1$.

- Let $\hat{i}', \hat{k}'$ be such that

$$\hat{k}' = \min\{k \in Z^+ \mid k = squad_{\hat{a}'}, \ k \geq squad_{\hat{a}},$$
$$\exists \hat{i}' \in I_{\hat{a}'} \text{ such that } \underline{inc}_{\hat{a}'\hat{y}'\hat{i}'} \leq APROC_{\hat{a}'\hat{y}'\hat{i}'} + k \leq \overline{inc}_{\hat{a}'\hat{y}'\hat{i}'}\}$$

(**Remark:** If $\hat{i}', \hat{k}'$ do not exist, then the move is infeasible)

- Assign the following components of x^m:

(a) Ship procurement and retirement: Same as in x^*.

(b) Aircraft procurement:

$$APROC_{ayi} := \begin{cases} APROC_{ayi} + \hat{k}', & \text{if } a = \hat{a}', y = \hat{y}', i = \hat{i}' \\ 0, & \text{if } a = \hat{a}', y = \hat{y}', i \neq \hat{i}' \\ \text{same as in } x^*, & \text{otherwise} \end{cases}$$

(c) Aircraft retirement:

$$ARET_{ay} := \begin{cases} ARET_{ay} - squad_{\hat{a}}, & \text{if } a = \hat{a}, y = \hat{y} \\ ARET_{ay} + squad_{\hat{a}}, & \text{if } a = \hat{a}, y = \hat{y}' \\ \text{same as in } x^*, & \text{otherwise} \end{cases}$$

- Assign the Ranking function as follows:

$$R_{JR}(x^m) = \begin{cases} 0, & \text{if } F(x^*) < F(x^m) \\ F(x^*) - F(x^m), & \text{otherwise} \end{cases}$$

6.3.2 Platform Exchange Deep Search Strategy

Exchange strategies refer to tentative moves involving the exchange of one platform procurement with another, but not necessarily in the same year, and not necessarily of the same type.

We have analyzed different ways to check for platform exchanges (Figures 6.8.a-d): In "Ship Exchange," we exchange the years that two different ships are procured; in "Aircraft Exchange," we exchange the years that two different aircraft are procured; in "Ship-Aircraft Exchange" we evaluate exchanges of a ship purchase with an aircraft purchase; and in "Plant Exchange" we exchange a ship purchase from a specific shipyard and year with the same ship class purchased from a different shipyard and/or year.
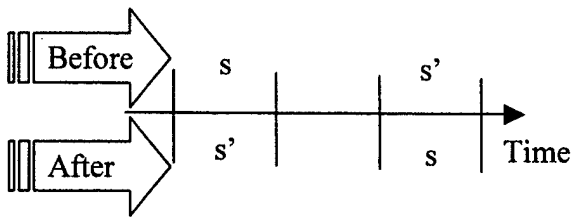
Figure 6.8.a: Ship Exchange.
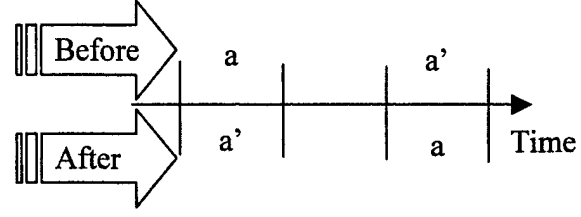
Figure 6.8.b: Aircraft Exchange.
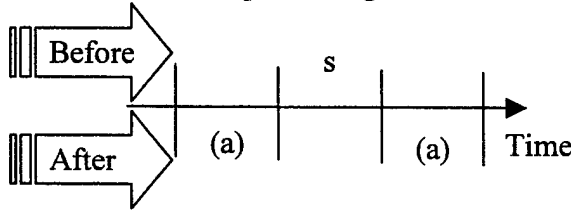
Figure 6.8.c: Ship-Aircraft Exchange.
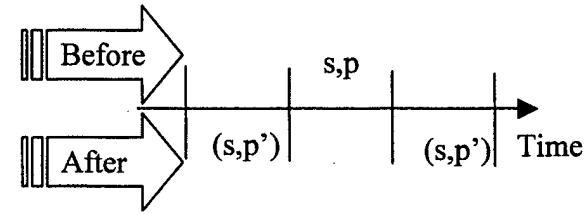
Figure 6.8.d: Ship-Plant Exchange.

Formal definitions are as follows:

1. Platform-Exchange (Ship part): Define the following new candidates, x^m, and their rankings, $R_{PES}(x^m)$, for $m=1,...,M$:

   - Select $m = \left(\hat{s}, \hat{p}, \hat{s}', \hat{p}', \hat{y}, \hat{y}'\right)$ where $\hat{s}, \hat{s}' \in S$, $\hat{p} \in P_{\hat{s}}$, $\hat{p}' \in P_{\hat{s}'}$, $\hat{y}, \hat{y}' \in Y$, $\exists \hat{q} \neq 0 \mid SPROC_{\hat{s}\hat{p}\hat{y}\hat{q}} = 1$, $\exists \hat{q}' \neq 0 \mid SPROC_{\hat{s}\hat{p}\hat{y}\hat{q}'} = 1$.

   - Assign the following components of x^m:

     (a) Aircraft procurement and retirement, and ship retirement: Same as in $\mathbf{x}^*$.

(b) Ship procurement:

$$SPROC_{spyq} := \begin{cases} 1, & \text{if } s = \hat{s}, p = \hat{p}, y = \hat{y}', SPROC_{spy',q-1} = 1 \\ 0, & \text{if } s = \hat{s}, p = \hat{p}, y = \hat{y}', SPROC_{spy'q} = 1 \\ 1, & \text{if } s = \hat{s}, p = \hat{p}, y = \hat{y}, SPROC_{spy,q+1} = 1 \\ 0, & \text{if } s = \hat{s}, p = \hat{p}, y = \hat{y}, SPROC_{spyq} = 1 \\ 1, & \text{if } s = \hat{s}', p = \hat{p}', y = \hat{y}, SPROC_{s'p'y,q-1} = 1 \\ 0, & \text{if } s = \hat{s}', p = \hat{p}', y = \hat{y}, SPROC_{s'p'yq} = 1 \\ 1, & \text{if } s = \hat{s}', p = \hat{p}', y = \hat{y}', SPROC_{s'p'y',q+1} = 1 \\ 0, & \text{if } s = \hat{s}', p = \hat{p}', y = \hat{y}', SPROC_{s'p'y'q} = 1 \\ \text{same as in } x^*, & \text{otherwise} \end{cases}$$

- Assign the Ranking function as follows:

$$R_{RES}\left(x^m\right) = \begin{cases} 0, & \text{if } F\left(x^*\right) < F\left(x^m\right) \\ F\left(x^*\right) - F\left(x^m\right), & \text{otherwise} \end{cases}$$

2. Platform-Exchange (Aircraft part), Platform-Exchange (Ship-Aircraft part), and Platform-Exchange (Plant part) are defined in a similar fashion.

6.4   Heuristic Lower Bound

Computing a good-quality lower bound (LB) by data inspection is not a trivial task. No feasible solution can have a better objective function value than the specified LB:

$$LB \leq F(x), \ \forall x \text{ feasible.}$$

CIPA constructs a lower bound by separately bounding each component of the separable objective function:

$$F(x) = F^{SM}(x) + F^{AM}(x) + F^{YB}(x) + F^{CB}(x) + F^{L}(x).$$

We will compute the following LBs: $LB^{SM} \leq F^{SM}(x)$, $LB^{AM} \leq F^{AM}(x)$, $LB^{YB} \leq F^{YB}(x)$, $LB^{CB} \leq F^{CB}(x)$, $LB^{L} \leq F^{L}(x)$, $\forall x$ feasible. Then, it is clear that:

$$LB = LB^{SM} + LB^{AM} + LB^{YB} + LB^{CB} + LB^{L} \leq F(x), \forall x \text{ feasible.}$$

We now describe how to calculate each of these individual LBs.

6.4.1 Lower Bound on Ship-Mission Penalty

The question is: "Can we establish any ship-mission shortfall by data inspection?" To be able to answer this question, we:

(a) Compute the actual maximum ship inventory for every ship class, s, and year y.

(b) Compute the minimum penalty incurred by ships due to lack of resources to accomplish the required Ship-Missions.

We must realize that part (a) above is not immediate. There are several factors conditioning the maximum possible inventory of a specific class s in year y:

- The initial inventory, $sinv_s^0$.

- The maximum inventory specified by the user, $\overline{sinv_s}$.
- The ongoing committed production, $csproc_{sy}$.
- The maximum and minimum procurement per year from each plant, $\overline{sproc_{spy}}$, $\underline{sproc_{spy}}$.

- The maximum total procurement from each plant, $\overline{stot_{sp}}$.
- The production and payment schedule at each plant (and, therefore, the earliest that ships can be acquired), SBb_{sp}, SCb_{sp}.

- And finally, the minimum yearly and cumulative retirements imposed by the user, $\underline{sret_{sy}}$, $\underline{csret_{sy}}$.

Each of the seven factors above may influence the maximum possible inventory of ships. Moreover, data for future years may influence the maximum inventory in earlier years. For example, meeting minimum cumulative retirements in the future may require retiring ships earlier, which in turn reduces the maximum inventory. The minimum procurement influences the maximum inventory: Because there is a maximum total procurement, if a minimum procurement exists in the future, then the maximum inventory in the present will be reduced. For example, if the maximum procurement over the time horizon is three ships and the minimum in the second year is one ship, then, in the first year, we cannot procure more than two ships.

Now, we present an overview of how the LB^{SM} bound is computed:

(1) Using the yearly and cumulative retirements, $\underline{sret_{sy}}$, $\underline{csret_{sy}}$, update the minimum cumulative retirement, $CUM\_SretMin(s,y)$:
$Cum\_SRetMin(s,y) := \max\{Cum\_SRetMin(s,y-1) + \underline{sret_{sy}}, \underline{csret_{sy}}\}$.

(2) Using the committed procurement, $csproc_{sy}$, calculate the cumulative committed procurements, $Cum\_CSProc(s,y)$:

$$Cum\_CSProc(s,y) = Cum\_CSProc(s,y-1) + \sum_p csproc_{spy}.$$

(3) Calculate the new maximum ship inventory, $Max\_SInv(s,y)$:

$$Max\_SInv(s,y) := \overline{sinv}_s - \left(sinv_s^0 + Cum\_CSProc(s,y) - Cum\_SRetMin(s,y) \right).$$

(4) Calculate (working backwards in time) the adjusted maximum ship inventory, $Adj\_Max\_SInv(s,y)$:

$$Adj\_Max\_SInv(s,y) := \min\{Max\_SInv(s,y), Adj\_Max\_SInv(s,y+1)\}.$$

(5) Calculate the maximum total ships to be procured, $Max\_Stot(s,p,y)$. Starting with $Max\_Stot(s,p,|Y|) = \overline{stot}_{sp}$, work backwards in time:

$$Max\_Stot(s,p,y) = Max\_Stot(s,p,y+1) - \underline{sproc}_{sp,y+1}.$$

(6) Calculate the initial maximum cumulative procurement that can be procured from each plant, $Ini\_Cum\_MaxProc(s,p,y)$:

$$Ini\_Cum\_MaxProc(s,p,y) = \sum_{1 \leq y' \leq y} \overline{sproc}_{spy'}.$$

and the first index $\hat{y}$ where this amount exceeds the maximum per plant: $\hat{y} = \min\{y \mid Ini\_Cum\_MaxProc(s,p,y) > Max\_STot(s,p,y)\}$ (or $\hat{y} = \infty$ if $Cum\_MaxProc(s,p,y) \leq Max\_STot(s,p,y), \forall y$)

(7) Calculate the adjusted maximum procurement, $MaxSProc(s,p,y)$:

$$MaxProc(s,p,y) := \begin{cases} 0, & \text{if } y < \max\{SBb_{sp}+1, SCb_{sp}+1\} \\ \overline{sproc}_{spy}, & \text{if } \max\{SBb_{sp}, SCb_{sp}\} \leq y < \hat{y} \\ Max\_STot(s,p,y) - Ini\_Cum\_MaxProc(s,p,y-1), & \text{if } y = \hat{y} \\ 0, & \text{if } y > \hat{y} \end{cases}.$$

(8) Accrue procurements from all plants, $All\_MaxSProc(s,y)$:

$$All\_MaxProc(s,y) := \sum_p MaxProc(s,p,y).$$

and calculate the cumulative amounts, $Cum\_All\_MaxSProc(s,y)$:

$$Cum\_All\_MaxProc(s,y) = \sum_{1 \leq y' \leq y} All\_MaxProc(s,y').$$

(9) Compare with the adjusted maximum ship inventory:

$$Cum\_All\_MaxProc(s,y) = \min\{Cum\_All\_MaxProc(s,y), Adj\_Max\_SInv(s,y)\}.$$

(10) Calculate an upper bound on the maximum inventory of ships:

$$UB\_SInv(s,y) := sinv_s^0 + Cum\_CSProc(s,y) - Cum\_SRetMin(s,y) +.$$
$$Cum\_All\_MaxProc(s,y)$$

(11) Calculate the inventory upper bound per Ship-Mission:

$$UB\_MInv(m,y) = \sum_{s \in S_m} UB\_SInv(s,y).$$

(12) Calculate the lower bound on the penalty per year and Ship-Mission:

$$LB(m,y) = \max\{0, smpen_m(\underline{smreq}_{my} - UB\_MInv(m,y))\};$$

and the total bound on Ship-Mission penalty:

$$LB^{SM} = \sum_{m \in M^S} \sum_{y \in Y} LB(m,y).$$

6.4.2 Lower Bound on Air-Mission Penalty

A lower bound on Air-Mission penalty can be obtained in an analogous fashion to the lower bound on Ship-Mission penalty. The differences in the procedure are summarized as follows:

(a)   There is no need to perform individual plant calculations.

(b)   The minimum year to produce an aircraft is $ABb_a + 1$, instead of $\max\{SBb_{sp} + 1, SCb_{sp} + 1\}$ as used in Step (7) of the ship procedure.

(c)   When calculating the maximum procurement per year, squadron sizes and segments for aircraft must be taken into account.

6.4.3 Lower Bound on Labor Penalty

A labor penalty arises when labor exceeds the maximum level or falls below the minimum level at any plant. We can derive a lower bound on these penalties.

This is an overview of how the LB^L bound is computed:

(1) Update the minimum and maximum amounts of ships that can be procured:

$$\underline{sproc}_{spy} = 0, \ \forall s \in S, p \in P_s; \forall y \le \max\{SBb_{sp}, SCb_{sp}\} - 1, \text{ and}$$
$$\underline{sproc}_{spy} = 0, \ \forall s \in S, p \in P_s; \forall y \ge |Y| + 1 - \max\{SBa_{sp}, SCa_{sp}\};$$

63

$$\overline{sproc}_{spy} = 0, \ \forall s \in S, p \in P_s; \forall y \leq \max\{SBb_{sp}, SCb_{sp}\} - 1 \text{ and}$$

$$\overline{sproc}_{spy} = 0, \ \forall s \in S, p \in P_s; \forall y \geq |Y| + 1 - \max\{SBa_{sp}, SCa_{sp}\}.$$

(2) Compute the yearly minimum and maximum possible labor per plant, Min_Labor(p,y), Max_Labor(p,y) according to production schedules and required minimum and maximum quantities, $\forall p \in P; \forall y \in Y$:

$$Min\_LABOR(p,y) = clabor_{py} +$$

$$\sum_{s \in S | p \in P_s} \ \sum_{\substack{y' \in Y | \\ y \leq y' \leq y + SCb_{sp}}} sworkb_{sp, \underline{sproc}_{spy'}, y' - y} +$$

$$\sum_{s \in S | p \in P_s} \ \sum_{\substack{y' \in Y | \\ y - SCa_{sp} \leq y' \leq y - 1}} sworka_{sp, \underline{sproc}_{spy'}, y - y'}$$

$$Max\_LABOR(p,y) = clabor_{py} +$$

$$\sum_{s \in S | p \in P_s} \ \sum_{\substack{y' \in Y | \\ y \leq y' \leq y + SCb_{sp}}} sworkb_{sp, \overline{sproc}_{spy'}, y' - y} +$$

$$\sum_{s \in S | p \in P_s} \ \sum_{\substack{y' \in Y | \\ y - SCa_{sp} \leq y' \leq y - 1}} sworka_{sp, \overline{sproc}_{spy'}, y - y'}.$$

(3) Compute the yearly penalties for violating the minimum and maximum limits per plant, $\forall p \in P; \forall y \in Y$:

$$LB^{L-}(p,y) = lpen_p^- \ \max\{0, \underline{pcap}_{py} - Max\_LABOR(p,y)\}$$

$$LB^{L+}(p,y) = lpen_p^+ \ \max\{0, Min\_LABOR(p,y) - \overline{pcap}_{py}\}.$$

(4) Compute a lower bound on labor cost due to under-employment, LB^{L-}, a lower bound on labor cost due to labor excess, LB^{L+}, and the total lower bound on labor cost, LB^L:

$$LB^{L-} = \sum_{y \in Y} \sum_{p \in P} LB^{L-}(p,y)$$

$$LB^{L+} = \sum_{y \in Y} \sum_{p \in P} LB^{L+}(p,y)$$

$$LB^L = LB^{L-} + LB^{L+}.$$

6.4.4 Lower Bound on Budget Penalty

A budget penalty applies when the expenditures in a given year exceed the maximum budget or fall below the minimum budget. Currently, a lower bound of zero is considered for both penalties, $LB^{B+} = 0$ and $LB^{B-} = 0$, respectively.

We could calculate lower bounds by analyzing

(a) the maximum possible procurement and O&M cost (to compute a better LB^{B-}), and

(b) the minimum possible procurement cost (to compute a better LB^{B+}).

However, it is unlikely that these bounds render non-zero values because

(a) it is clear that we will be able, in general, to expend more than the minimum budget, and

(b) we do not expect that the minimum feasible purchase already exceeds the maximum budget, since that problem would be unrealistic under the present conditions.

However, we continue to seek other bounds for missions, labor and budget, as well as bounds that do not rely on individually bounding each of these penalties.

7. Features of the Exact Algorithm

Exact Solver (ES) implements a simplified version of the CIPA model in GAMS, and then uses commercial optimization software (e.g., CPLEX, OSL) to solve it. The ES also produces a lower bound on the optimal solution of the problem.

In this section, we detail the calculation of the lower and upper bounds through the ES.

7.1   Lower Bound

Any relaxation of the CIPA model constraints produces a new model whose optimal solution is no worse (i.e., it has objective function value no greater) than the optimal solution of the original CIPA model. The goal is to find a relaxed model that is easy to solve and yields a good bound (i.e., close to the optimal solution of the original problem).

We compute the so-called "exact lower bound" by relaxing all integrality restrictions in the CIPA model. In other words, the constraints:

$$APROC_{ayi} \in Z^+, \ ARET_{ay} \in Z^+, \ AP_{ayi} \in \{0,1\}, \ SPROC_{spyq} \in \{0,1\}, \ SRET_{sy} \in Z^+$$

become

$$APROC_{ayi} \geq 0, \ ARET_{ay} \geq 0, \ 0 \leq AP_{ayi} \leq 1, \ 0 \leq SPROC_{spyq} \leq 1, \ SRET_{sy} \geq 0.$$

In addition to these changes, we also disregard the squadron size requirement for aircraft procurement (3.46).

This relaxation is much easier to solve, taking a minute or two for the largest cases tested. This is much longer than the fraction of a second required to compute a heuristic lower bound. The extra time typically provides a better lower bound.

7.2   Upper Bound

7.2.1 The Simplified Model

The ES CIPA model can be solved, but we cannot guarantee that it can always be solved in a reasonable amount of time. Even when we find an admissible solution, we cannot guarantee that we can find a quantitative assessment of solution quality (lower bound) arbitrarily close to the cost of the incumbent solution. State-of-the-art mathematical programming techniques to solve an integer linear model like CIPA entail (in the worst case) an exponential number of operations to produce a strictly optimal solution to the problem.

To reduce the computational burden of these algorithms and expedite the "branch-and-bound" search, we simplify CIPA by relaxing integrality requirements for

66

aircraft procurement and retirement, and for ship retirement. That is, the stipulations in CIPA:

$$APROC_{ayi} \in Z^+, \; ARET_{ay} \in Z^+, \; AP_{ayi} \in \{0,1\}, \; SPROC_{spyq} \in \{0,1\}, \; SRET_{sy} \in Z^+$$

are relaxed to

$$APROC_{ayi} \geq 0, \; ARET_{ay} \geq 0, \; AP_{ayi} \in \{0,1\}, \; SPROC_{spyq} \in \{0,1\}, \; SRET_{sy} \geq 0.$$

We also disregard the squadron size requirement for aircraft procurement (3.46), which in turn reduces the number of segments and binary $AP_{ayi} \in \{0, 1\}$ variables.

This simplified CIPA model is a relaxation, but a stronger one than that used to compute the exact lower bound. Accordingly, the optimal solution can be expected to be a stronger lower bound.

The principal disadvantage of adopting this simplified model is the (likely) loss of integer feasibility. This entails dealing with a solution that possibly contains fractional values for the retirement of ships and aircraft and the procurement of aircraft, besides failing to meet the squadron size production requirement. We have devised a post-processor to heuristically round the fractional integer variables in the simplified model solution to a nearby integer solution. The remainder of this section presents this process.

7.2.2 Rounding Post-Process

Each ship or aircraft retirement, $SRET_{sy}$ or $ARET_{ay}$, is easily rounded to the nearest integer, $R(SRET_{sy})$ and $R(ARET_{ay})$, respectively, where the "round" function R(x) is defined as follows:

$$R(x) = \begin{cases} [x]+1, & \text{if } x - [x] \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

An aircraft procurement, $APROC_{ayi}$, is also rounded to the nearest integer but, in addition, we must observe the squadron size conditions. While doing this, we need to ensure that:

- The new $APROC_{ayi}$ is a multiple of $squad_a$.
- The new $APROC_{ayi}$ is within the limits of segment i, $\underline{inc}_{ayi}, \overline{inc}_{ayi}$, for some $i \in I_a$.
- We do not exceed the minimum and maximum yearly procurement, $\underline{aproc}_{ay}, \overline{aproc}_{ay}$.
- We do not exceed the maximum total procurement, $\overline{atot}_a$.

67

- We do not exceed the maximum inventory $\overline{ainv_a}$.

The rounding heuristic is as follows:

1. Let $SPROC_{sy}, SRET_{sy}, APROC_{ayi}, ARET_{ay}$ be the solution to the relaxed problem. Assign:

$$\begin{cases} R\_SRET_{sy} := R(SRET_{sy}), \forall s \in S, y \in Y \\ R\_ARET_{ay} := R(ARET_{ay}), \forall a \in A, y \in Y \\ R\_APROC_{ayi} := R(APROC_{ayi}), \forall a \in A, y \in Y, i \in I_a \end{cases}.$$

2. Assign the current aircraft configuration:

$$\begin{cases} APROC_{ayi} := R\_APROC_{ayi}, \forall a \in A, y \in Y, i \in I_a \\ ARET_{ay} := R\_ARET_{ay}, \forall a \in A, y \in Y \end{cases}.$$

3. Set $a:=1$, $y:=1$, $i \mid AP_{ayi} = 1$ (note that, given a and y, by eq. (3.4) there is only one segment i verifying $AP_{ayi} = 1$); SOLUTION:= "NO."

4. Find the minimum k^* such that:

$$\begin{cases} R\_APROC_{ayi} \leq k^* \leq \overline{aproc}_{ay}, \\ \underline{inc}_{ayi'} \leq k^* \leq \overline{inc}_{ayi'} \text{ for some } i' \in I_a, \text{and.} \\ k^* \text{ is a multiple of } squad_a \end{cases}$$

(k^* is the nearest integer- and squadron-size-feasible solution closest to the original $APROC_{ayi}$ by above, but it may fall in another segment i'.)

5. If k^* in Step 4 does not exist, proceed to Step 8.

6. Assign a new $APROC_{ayi'} := k^*$ and compute the new total procurement and inventory levels:

$$\begin{cases} TotProc_a = \sum_{y \in Y} \sum_{i \in I_a} APROC_{ayi} \quad \text{and} \\ AINV_{ay} = ainv_a^0 + \sum_{y' \in Y | y' \leq y} caproc_{ay'} + \sum_{y' \in Y | y' \leq y} \sum_{i \in I_a} APROC_{ay'i} - \sum_{y' \in Y | y' \leq y-1} ARET_{ay'} \end{cases}$$

7. Are $TotProc_a \leq \overline{atot}_a$ and $AINV_{ay} \leq \overline{ainv}_a$? If so, proceed to Step 10.

8. Find the maximum k^* such that:

$$\begin{cases} \underline{aproc}_{ay} \leq k^* \leq R\_APROC_{ayi}, \\ \underline{inc}_{ayi'} \leq k^* \leq \overline{inc}_{ayi'} \text{ for some } i' \in I_a, \text{ and.} \\ k^* \text{ is a multiple of } squad_a \end{cases}$$

(k^* is the nearest integer- and squadron-size-feasible solution closest to the original $APROC_{ayi}$ by below, but it may fall in another segment i'.)

9. If k^* in Step 8 does not exist, proceed to Step 15.

10. Assign the new $APROC_{ayi'} := k^*$.

11. Increase y by 1.

12. If $y > |Y|$, then increase a by 1 and set $y: = 1$.

13. If $a > |A|$, then set SOLUTION: ="YES" and proceed to Step 15.

14. Return to Step 4.

15. If SOLUTION="Yes," then the rounded solution is as follows:

$$\begin{cases} SPROC_{spy} := SPROC_{spy}, \forall s \in S, p \in P_s, y \in Y \\ SRET_{sy} := R\_SRET, \forall s \in S, y \in Y \\ ARET_{ay} := R\_ARET_{ay}, \forall a \in A, y \in Y \\ APROC_{ayi} := APROC_{ayi}, \forall a \in A, y \in Y, i \in I_a \end{cases}$$

Otherwise, we find no integer solution to the problem.

After the rounded solution is computed, the main decision variables (as they appear in Step 15) are fixed in the CIPA model. We solve this restricted model again in order to fix the remaining control variables. The final ES solution is then returned to the Solver, where it is checked for feasibility and objective function value.

8. CIPA Results

8.1   Data Used for Testing CIPA

We have assessed CIPA with a number of scenarios created from a realistic baseline case after Baran (2000). This baseline has 45 ship classes, 30 aircraft types, 11 production facilities, 17 Ship-Missions, and 12 Air-Missions over a 30-year planning horizon (FY01 to FY30, of which the initial FY01 to FY05 are frozen by Program Objective Memorandum (POM)). This case derives principally from the U.S. Naval Center for Cost Analysis.

Ship-Mission Areas	Associated Ship Classes
Destroyers	FFG, DDG, DDGX, DD, DD21
Cruisers	CG, CG21
Carriers	CVN63, CVN65, CVN68
Attack Submarines	SSN774, SSN688, SSN21
Strategic Missile Submarines	SSBN726, SSBNX
Amphibious Assault Ships	LHA, LHD, LHX
Landing Dock Ships	LSD36, LSD41
Amphibious Transport Ships	LPD4, LPD17
Mine Countermeasure	MCM1, MCMX
Mine Hunter Ships	MHC50, MHCX
Command Ships	LCC19
Logistic AO ships	AO187, TOAX
Logistic AOE Ships	AOE1, AOE6, TADCX
Support AS Ships	AS39, ASX
Support ARS Ships	ARS50, ARSX
Support ATF Ships	ATF166, ATFX
Support TAGOS Ships	TAGOS1, TAGOS19, TAGOS23

Table 1. Baseline case: Ship-Mission areas and associated ships. After Baran (2000).

Tables 1 and 2 summarize ship-mission and air-mission areas, respectively. A few ship classes and aircraft types with no future programs, and thus no degrees of freedom (such as LST-1179, MCS-12, etc. for ships, and F-5EF, EA-6, etc. for aircraft), have been intentionally removed. The economic impact of these now-exogenous programs is reflected by the "other cost" mechanism in CIPA.

Air-Mission Areas	Associated Aircraft Types
Fighter Aircraft	JSFN, JSFMC, F18EF, F18AB, F18CD, F14, AV8B
Attack Aircraft	EA6B, F18G
ASW Aircraft Group 1	S3B, CSAASW
ASW Aircraft Group 2	P3C, MMA
Early Warning Aircraft	E2C, E2X
Transport Aircraft	C2AB, C2X
Utility Aircraft	C12, UCX
Training Aircraft Group 1	T44, METX
Training Aircraft Group 2	T45, JTTX
Training Aircraft Group 3	T34, JPATS
Rotary Wing Group 1	TH57, THX
Rotary Wing Group 2	MV22, CH46E, CH53D

Table 2. Baseline case: Air-Mission areas and associated aircraft. After Baran (2000).

Table 3 shows the shipyards considered in our test cases and the ship types that can be built in each. Note that the same ship type can be produced at different shipyards and (possibly) different production rates and costs will apply at each.

Shipyard	Ships Produced
Bath	DDG, DDGX, DD21,CG21
Ingals	DDG, DDGX, DD21,CG21, LHX
News	CVN68, CVX, SSN774, SSNX, SSBNX, LCCX
Eboat	SSN774, SSNX, SSBNX
Avon	LSDX, LPD17, TAOX
Peterson	MCMX, ARSX
Interm	MHCX
Phil	LCCX
NationalS	TADCX
Locheed	ASX
Marinette	TATFX

Table 3. Baseline case: Shipyards and ships produced. After Baran (2000).

The minimum and maximum annual budgets in the baseline case are respectively about $35 billion and $51 billion, and are expressed as a cumulative restriction over the planning horizon. Specific details regarding other data (such as production rates and costs, O&M costs, mission requirements, industry work-force levels, etc.) can be found in Baran (2000).

8.2 Output Analysis From the Solver

CIPA is operated from its GUI. From the GUI, all the necessary data files are presented to the Solver, and each solution is retrieved and presented, making the optimization process easy and transparent to a planner.

71

An experienced planner might also be able to manually create data files, run the optimization and analyze the results. A very detailed example of input files required by the solver and the resulting output can be found Appendix A. The remainder of this section describes the solver output, called "CIPA.log."

The log file summarizes solver results. This file either contains the heuristic and exact solver findings, or if the solver has failed, a diagnosis of the failure (e.g., inability to find data files, inconsistent data found, etc.).

Figures 8.1–8.3 are specimens from CIPA.log files:

> *Initialzing Parameters...*
> *Checking Folders...*
> *Reading Data for case...*
> ... *Case_1_1: From Case_1_0 increasing mission requirements by 10%*
> *Optimizing...*
> *Writing Gams Data...*

Figure 8.1: CIPA.log (initialization).

The Solver is initialized by setting some parameters and verifying availability of essential folders and files. After data have been read and checked for consistency, optimization starts. Some files are created for the exact (GAMS) solver.

> **... LB heuristic: 695028.6**
> ... Gams RMIP invoked. Waiting for termination...
> ... ...Gams RMIP done. Op.Sys. status=   0
> **... LB gams: 718213.1**
> **... Searching for an Initial Solution...**
> ... ... Heuristic initial solution...
> ... ... Checking feasibility...
> ... ... Feasible solution.
> ... ... Updating variables and objective
> **... ... Initial Solution process finished. Cost: F= 1.2725737E+07**

Figure 8.2: CIPA.log (lower bound and initial solution).

We see the heuristic lower bound, the "exact" lower bound, and then confirmation that the heuristic search for a feasible solution has succeeded, resulting in an objective function value of 12,725,737.

72

```
... Searching for Retirement Improvements
... ... Ship Retirement Changed.  Cost F=  1.2726499E+07
... ... Ship Retirement Changed.  Cost F=  1.2727262E+07
... Searching for Mission Improvements
... ... Ship-Mission improvement.  Cost F=  1.2377644E+07
... ... Ship-Mission improvement.  Cost F=  1.2074045E+07

            (DELETED TEXT)
... ... Air-Mission improvement.  Cost F=  1.0786315E+07
... ... Ship-Mission improvement.  Cost F=  1.0649005E+07
... ... Ship-Mission improvement.  Cost F=  1.0513149E+07
... ... Air-Mission improvement.  Cost F=  1.0298276E+07

            (DELETED TEXT)
... ... Air-Mission improvement.  Cost F=  735899.9
... Searching for Retirement Improvements
... Searching for Labor Improvements
... ... Labor improvement.  Cost F=  734709.2
... ... Labor improvement.  Cost F=  733584.1

            (DELETED TEXT)
... ... Labor improvement.  Cost F=. 731962.2
... Searching for Retirement Improvements
... Searching for Budget Improvements
... Searching for Retirement Improvements
... Deep Local Search
... ... Ship Ret-Proc joint move
... ... Air Ret-Proc joint move
... ... Ship-Exchange joint move
... ... ... Aircraft Exchanged.  Cost F=  731885.0
... ... Ship-Air Proc. exchange joint move
... ... Plant-Exchange joint move
... ... ... Plant-Years Exchanged.  Cost F=  730260.6
... Saving Heuristic solution
```

Figure 8.3: CIPA.log (Heuristic Solver).

The heuristic starts by reconfiguring some retirements, even if they do not improve the total objective function. Next, the mission strategy searches for improvements by adding new ships and aircraft to our plan. Then, we search for better platform retirements, labor penalty reduction, platform retirements again, budget penalty reduction and, once more, platform retirements. Finally, we do deep-search a final best heuristic solution that turns out to cost $730,260.60.

```
... Gams MIP invoked. Waiting for termination...
... ...Gams MIP done. Op.Sys. status=      0
... Gams solution obj.:   787486.6
Gams soln. read
... Checking Gams Solution...
... ... Checking feasibility...
... ... Feasible solution.
... ... Updating variables and objective
... ... Gams Solution feasible : Cost=   787486.6
... ... valid solution (costs match)
... Saving Gams solution
... Restoring best Solution...
... ... (LB=Gams) ·
... ... (UB=Heuristic)
... Restoring Heur solution
... ... Updating variables and objective
```

Figure 8.4: CIPA.log (Exact Solver and best solution).

Here, we optionally seek an exact solution by committing some allotted time to a GAMS mixed-integer solver. We retrieve the best "exact" incumbent solution found—in this case its cost is $787,486.60—and compare and report the best lower and upper bound from both solvers. In this case, the best lower bound is provided by the Exact Solver, but the best upper bound (feasible solution) is provided by the Heuristic Solver.

74

```
RESULTS FOR CASE:
    Case_1_1: From Case_1_0 increasing mission requirements by 10

Program Status: 1      (Program finished correctly)


Solution Status: 2     (Feasible solution)


HEURISTIC SOLVER SUMMARY:
Penalty type            Value (UB)      (Lower Bound)
-------------------     -----------     --------------
Budget: F_B                   0.00     (         0.00)
Cum. Budget: F_CumB           0.00     (         0.00)
Labor: F_L                59344.12     (     36867.58)
Ship-Missions: F_SM      518114.41     (    518114.25)
Air-Missions: F_AM       152802.00     (    140046.80)
Total: F                 730260.56     (    695028.63)


GAMS SOLVER SUMMARY:
Penalty type            Value (UB)      (Lower Bound)
-------------------     -----------     --------------
Budget: F_B                   0.00     (not computed)
Cum. Budget: F_CumB           0.00     (not computed)
Labor: F_L                54526.24     (not computed)
Ship-Missions: F_SM      532908.13     (not computed)
Air-Missions: F_AM       200052.16     (not computed)
Total: F                 787486.63     (    718213.06)


OVERALL ALGORITHM SUMMARY:
Penalty type             Value          (Lower Bound)
-------------------     -----------     --------------
Total: F                 730260.56     (    718213.06)
```

Figure 8.5: CIPA.log (Results summary).

CIPA reports the status of the execution (program and solution), which indicates that the optimization was carried out successfully. There is a report for the heuristic solver and (optionally) one for the exact solver, both itemized by category of penalty. The overall summary shows the final solution ($730,260.56) and lower bound ($718,213.06).

```
Time initializing parameters                          0.35
Time reading user data                                0.56
Time writing gams data                                1.00
Time optimizing                                     242.36
        (Lower Bound)                    (          62.66)
        (Initial Solution)               (           0.07)
        (SMissions)                      (          24.78)
        (AMissions)                      (           8.29)
        (Labors)                         (           0.94)
        (SBudgets)                       (           0.00)
        (ABudgets)                       (           0.00)
        (SRetirements)                   (           0.29)
        (ARetirements)                   (           0.14)
        (Deep_Search)                    (          12.20)
        (Gams UB)                        (         131.81)
        (Restore_Best)                   (           0.07)
Time printing results:                                0.31

Total Time CIPA:                                    243.59
```

Figure 8.6: CIPA.log (Time report).

Time report itemized by category. The optimization time is broken down into the different strategies used.

8.3 Comparison Between Heuristic and Exact Solver

We have implemented the Solver module in a 1 GHz personal computer with a Pentium III© processor and 1 GB of RAM, under the operating system Windows 2000© [2002].

The Exact Solver implements the CIPA model in GAMS modeling language [Brooke et al. 1996] and solves it by using the OSL [GAMS/OSL 2002] or CPLEX [GAMS/CPLEX 2002] optimization libraries. The Heuristic Solver has been implemented in Fortran [Digital Visual Fortran 1998].

Table 4 shows a comparison of performance between the HS and ES (with GAMS/CPLEX) in 24 cases created as excursions from the baseline case (identified as Case 1.0 in that table). The excursions differ from each other by ·

(a) whether a yearly budget (YB) and/or a cumulative budget (CB) are considered or not,

(b) the mission requirement increment (MRI) from the baseline case, and

(c) the budget increment (BI) from the baseline case.

We have explored combinations of these factors for MRIs equal to -15%, 0%, 10%, and 25%, and BIs equal to -20% and 0%.

76

As expected, the LB computations are clearly superior for the exact procedure, which gives an idea of the difficulty in coming up with non-trivial lower bounds by simple examination of the data. The heuristic bound can be computed in less than a second, whereas the exact bound needs between one and two minutes using GAMS/CPLEX, and 20% more on average with GAMS/OSL.

The following two columns (headed "Exact Solution" and "Heuristic Solution," respectively) show the findings by the exact and heuristic methods. The exact method uses GAMS/CPLEX and the figures indicate the best solution obtained after 10 minutes of computation. This is a hard integer linear program, and no case is solved during the allotted time. About half of these cases do not even yield a feasible solution. We ran these cases for hours and some of them are essentially intractable. In contrast, the heuristic solver seems to perform reasonably. The computation time for the heuristic is about 30 seconds in each of the runs, yielding high quality per unit time.

Analyzing the results in Table 4 (as well as a lot more computational experience not shown), we find the heuristic solver highly effective and recommend it. We also find it useful to calculate an exact lower bound to support the value and accuracy of the heuristic solution. But, we only recommend the use of the exact solver to calculate a feasible solution if either:

(a) the problem dimension is small, or

(b) the heuristic solution proves unreasonable, or

(c) the heuristic solution gap is very high after computing the exact lower bound.

In any case, given the high volatility of the ES computational time, we recommend enforcing a maximum limit (e.g., one or two hours).

77

					Heuristic	Exact	Exact	Heuristic	Gap	Gap
Case	YB	CB	MRI	BI	LB	LB	Solution	Solution	(Exact)	(Heuristic)
1.0		X			102,245	119,577	?	124,565	?	4.17
1.1		X	10%		695,028	718,213	787,486	736,652	9.65	2.57
1.2		X	25%		1,732,651	1,776,227	1,885,502	1,854,386	6.15	4.40
1.3		X	-15%		60,374	72,953	?	78,225	?	7.23
2.0		X		-20%	102,245	119,577	?	127,044	?	6.24
2.1		X	10%	-20%	695,028	734,973	837,348	867,633	13.93	18.05
2.2		X	25%	-20%	1,732,651	2,053,046	2,265,498	2,246,046	10.35	9.40
2.3		X	-15%	-20%	60,374	72,953	128,390	76,798	75.99	5.27
3.0	X	X			102,245	150,476	?	179,071	?	19.00
3.1	X	X	10%		695,028	750,519	873,864	793,604	16.43	5.74
3.2	X	X	25%		1,732,651	1,830,805	2,001,855	1,926,352	9.34	5.22
3.3	X	X	-15%		60,374	103,791	?	122,791	?	18.31
4.0	X	X		-20%	102,245	135,488	?	145,124	?	7.11
4.1	X	X	10%	-20%	695,028	751,643	907,631	886,261	20.75	17.91
4.2	X	X	25%	-20%	1,732,651	2,074,060	2,303,384	2,343,700	11.06	13.00
4.3	X	X	-15%	-20%	60,374	88,810	?	95,945	?	8.03
5.0	X				102,245	150,476	?	176,455	?	17.26
5.1	X		10%		695,028	750,519	835,994	787,605	11.39	4.94
5.2	X		25%		1,732,651	1,830,805	1,980,475	1,922,182	8.18	4.99
5.3	X		-15%		60,374	103,791	?	119,673	?	15.30
6.0	X			-20%	102,245	135,488	?	143,235	?	5.72
6.1	X		10%	-20%	695,028	741,907	854,843	867,600	15.22	16.94
6.2	X		25%	-20%	1,732,651	1,854,925	2,022,075	2,218,208	9.01	19.58
6.3	X		-15%	-20%	60,374	88,810	?	98,216	?	10.59

Table 4. Test cases run with the CIPA ES and the CIPA HS.

Legend: **YB:** Yearly budget; **CB:** Cumulative budget; **MRI:** Mission requirement increment (from baseline case); **BI:** Budget increment (from baseline case); Heur LB: Heuristic lower bound; **Exact LB:** Exact lower bound calculated with GAMS/CPLEX; **Exact Solution:** Exact solution calculated with GAMS/CPLEX in a maximum of 10 min; **Heuristic Solution:** Heuristic solution; **Gap (Exact):** Max. relative gap (%) for the exact solution; **Gap (Heuristic):** Max. relative gap (%) for the heuristic solution.

9. CIPA Project Contributions, Deliverables, and Current Status

9.1   Contributions

CIPA is being developed in the Operations Research (OR) Department at the Naval Postgraduate School (NPS), Monterey, CA. Since 1999, CIPA has been funded by the Chief of Naval Operations, Assessment Division (N81) and the Office of Naval Research.

CIPA principal investigators are Distinguished Professor Gerald Brown and Associate Professor Robert Dell. Research Assistant Professors Javier Salmeron and Anton Rowe have developed and integrated the CIPA algorithmic procedures and GUI, respectively.

A number of NPS OR graduate students have contributed to CIPA through the following Masters Theses:

Lt. R. J. Field (U.S. Navy): "Planning Capital Investment in Navy Forces," December 1999.

Lt. N. Baran (Turkish Navy): "Optimizing Procurement Planning of Navy Ships and Aircraft," December 2000.

LCDR R. M. Garcia (U.S. Navy): "Optimized Procurement and Retirement Planning of Navy Ships and Aircraft," December 2001.

9.2   Deliverables

Official versions of CIPA are those that have been delivered to N81 as testing prototypes or final versions.

Versions are coded as follows: x.yy.zz where:
x: p indicates prototype version, d indicates developing version, and w indicates working version
yy: consists of two numbers indicating the interface version
zz: consists of two numbers indicating the heuristic solver version

Table 9.1 shows those versions that have been delivered as of May 1, 2002.

Version	Date Delivered	Solver	GUI
P.01.01	02/28/01	Only heuristic algorithm.	Not scalable. Field's data preloaded. No documentation.
P.03.03	06/04/01	Only heuristic algorithm.	Not scalable. Field's data preloaded. No documentation.
P.07.04	11/13/01	Only heuristic algorithm.	Not scalable. Field's data preloaded. GUI tour [CIPA Quick Tour, 2001]. Basic GUI user's guide [CIPA: User's manual, 2001] (unfinished).
P.08.05	Internal use only	Only heuristic algorithm. Effectiveness not included (see Appendix B)	Not scalable. Effectiveness data is not included. Instead, a one-to-one mapping is assumed: platforms rated 0, 1 or 2 do not accomplish the mission at all, platforms rated 3, 4 or 5 do accomplish the mission entirely. Same documentation as P.07.04.
V.08.27	May 2002	Only heuristic algorithm	Not scalable. Effectiveness data is included.

Table 9.1. Official versions.

9.3 Other Documents

Table 9.2 summarizes the documentation associated with the software development of CIPA as well as other manuals.

Document (File)	Purpose
CIPA General Report* (This same document.)	To present CIPA features, including the mathematical model, an introduction to the GUI, the heuristic and exact solvers, computational results, etc.
Optimization Model (CIPA_Model.doc)	To state the optimization model of the problem.
Data Structure (Data_Structure.doc)	To describe all the input and output (as well as relevant throughput) for the solver and the interface. This includes a description of the solver input and output files.
Cipa Solver Versions (CIPA Solver Versions .doc)	To describe the changes in the different versions of the CIPA algorithm. This may be due to enhancements of the existing procedures, additional functional requests, etc. The document also explains how the optimization model, data structures, etc. need to be modified to accommodate these changes.
Hierarchical Diagram (Hierarchical_diagram.ppt)	To represent the hierarchical structure of the procedures implemented in the heuristic solver.
Explanation of the Objective Function (Obj_Func.ppt)	To show a graphical explanation of the objective function components, as well as a hierarchical diagram to explain the update procedures involved as the solution is modified.
Heuristic Procedures Overview** (Heur_Procedures.doc)	To describe the main features of the routines coded for the Heuristic solver: name, purpose, level, called by, I/O arguments, internal and external calls, and other details.
Exact Solver Scheme (GAMS_Framework.ppt)	To represent the hierarchical structure of the procedures implemented in the Exact Solver.
Interface Data Validation*** (Data_checklist.doc)	To describe the necessary validations required in a future fully scalable user's interface. It contains description of special calculus and other validation procedures (also called "triggers") to be made as the planner enters data into the system (i.e., during the "Add," "Delete," and "Modify" methods).
Tour (Tour.pps)	To show a quick tour through the CIPA GUI.
User's Manual (Manual.doc)	To describe all the CIPA GUI features for a generic planner: entering data, running the optimization model, understanding solution charts and reports, etc.

Table 9.2.

* This document consolidates the prototype version P.07.04.
** In preparation.
*** Document not up-to-date. It is contingent upon the creation of a fully scalable application.

9.4   Current Status of the CIPA Project

As of August 1, 2002, the last version delivered to N81 is V.07.28.

In addition to consolidated versions of the CIPA system, independent research is devoted to accommodate potential user requirements. Some enhanced versions of the model and solver that have not been contemplated yet in any of the official versions are described in Appendix B of this document.

Additional research, such as aircraft age management by LCDR R. M. Garcia [2001] has not been incorporated into the existing model, algorithms, and GUI in either official or in-progress versions of CIPA.

Ongoing work by other OR NPS students is focused on improving the solution time of the exact solver by employing integer partition schemes, as well accounting for end-effects. For the future, we consider a stochastic formulation of the model to deal with uncertain budget and mission requirements.

Appendix A: Data Structure

A.1   Entities

The following main entities will be used to describe the Data Model:

Entity	Shortwave
Year	Y
Ship	S
Aircraft	A
Ship-Mission	SM
Air-Mission	AM
Plant	P
Quantity	Q
Payment Year (before)	L
Payment Year (after)	L
Construction Year (before)	N
Construction Year (after)	N
Miscellaneous	G

Other derived entities appear as the result of relationships between the main ones. For example, there is a relationship: "An aircraft can be delivered in one or many years" and "one year can receive one or many aircraft." For data consistency, we create the entity "AY", in order to split the "many-to-many" into two one-to-many relationships. This analysis allows us to identify the different elements of the problem and their relationships, and what elements may or mau not coexist with others. Figure A.1 shows the complete entity-relation diagram (ERD) for CIPA.

A.2   Tables of Data

Following the ERD, we present the tables required by CIPA.

We indicate in each table:

"Key": Those fields that are primary or foreign keys will be marked as "k", otherwise we leave them blank.
>   In addition, the heuristic algorithm has a parallel identification with ordinal numbers besides the codes used in the interface and database. We will indicate those fields as "k(H)" (keys used only by the heuristic).

"Field": Field name
"Description": Field description
"T": Field type:
>   "I", integer number
>   "R", real number
>   "L", logic (Boolean): 0=No, 1=Yes
>   "An", Alphanumeric of length n

"Or.": Data origin:

"Dat", raw data provided by the user to be employed in both the interface and the algorithm

"Dat (I)", raw data provided by the user to be used only in the interface

"F", fixed data (the user cannot view or modify it) required by the heuristic. Its specific fixed value is indicated in the "Remarks" column.

"Cal", data calculated in the interface with raw data from the user and employed in both the interface and the algorithm

"Cal (I)", Data calculated in the interface using raw data from the user. They are used in the interface only

"Cal (H)", Data calculated in the heuristic using raw data from the user. They are used in the heuristic only

"Ctr", Control data for the heuristic search. For the interface they are treated as Fixed (F) data

"Res", Result from the algorithm

"Aux", Auxiliary information from the algorithm

"Model": Specifies the equivalence in the model formulation (parameter, variable, set or index), see "CIPA_Model.doc". If not specified, we will use:

"Cal", to indicate that the model has not a explicit parameter or variable associated to the field but it can be calculated by doing some calculation with existing data or variables in the model .

"NA", to indicate that the model does not use that specific data or results

"Remarks": Describes any other information of interest.

CIPA : Entity-Relation diagram

AM

SM

AAM AMY General (G) SMY SSM

A AY Y SY S

PY

AYI P

SPY SP

one —to→ many

Dashed lines refer to
pseudo-relationships or
pseudo-entity caused by the
value of an attribute in
another entity
Grey entities generate
primary and foreign keys

Q L Ll N Nn

SPL SPLl SPN SPNn

SPQL SPQLl SPQN SPQNn

I

AY

SY
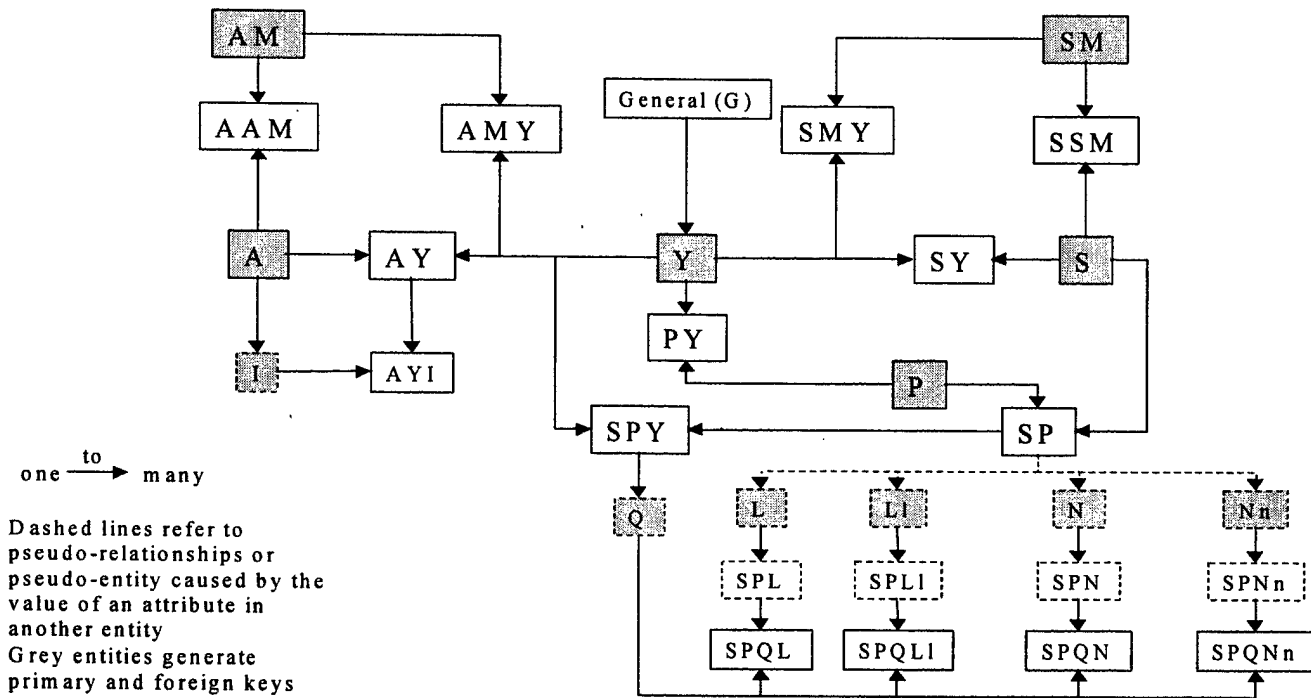
Figure A.1: Entity-relation diagram.

Table "General"

Key	Field	Description	Type	Origin	Model	Remarks
	Plan_Code	Code of the current plan or case under analysis	A128	Dat	(NA)	
	Year_Ini	Initial year of the case	I	Dat (I)	I	
	Year_End	Final year of the case	I	Dat (I)	$\lfloor Y \rfloor$	
	frac	Historical fraction of ship costs	R	Dat	$frac$	
	apn5	Historical fraction of aircraft costs	R	Dat	apn_5	
	ISol_User	Whether the user is providing an initial solution or not	L	F	(NA)	0 (= "No") for the interface
	Gams_Opt	Use GAMS during the optimization	I	Dat	(NA)	0=No 1=Only for LB 2=Yes (LB & UB)
	F_B	Total budget penalty	R	Res	(Cal)	Update when F_B_y(y) changes
	F_CumB	Total cumulative budget penalty	R	Res	(Cal)	Update when F_CumB_y(y) changes
	F_L	Total labor penalty	R	Res	(Cal)	Update when F_L_y(y) changes
	F_SM	Total Ship-Mission penalty	R	Res	(Cal)	Update when F_SM_y(y) changes
	F_AM	Total Air-Mission penalty	R	Res	(Cal)	Update when F_AM_y(y) changes
	F	Total penalty	R	Res	F	Update when F_y(y) changes
	LB_F_B	Lower bound on total budget penalty	R	Res	(Cal)	
	LB_F_CumB	Lower bound on total cumulative budget penalty	R	Res	(Cal)	
	LB_F_L	Lower bound on total labor penalty	R	Res	(Cal)	
	LB_F_SM	Lower bound on total ship mission penalty	R	Res	(Cal)	
	LB_F_AM	Lower bound on total air mission penalty	R	Res	(Cal)	
	LB_F	Lower bound on total penalty	R	Res	(Cal)	Update when LB_F_B, LB_F_CumB, LB_F_L, LB_F_SM or LB_F_AM change
	CIPA_Time	Total Computational time	R	Aux	(NA)	
	Error_Code	Error Code	I	Aux	(NA)	
	Error_Msg	Error Message	A128	Aux	(NA)	
	Error_Line	Error Line in case it occurs in a data file	I	Aux	(NA)	0 if unavailable
	Line_Header	Line Header in case the error occurs in a data file	A128	Aux	(NA)	Blank if any
	Prog_Status	Program Status Code	I	Aux	(NA)	1: Finished correctly 2: Error
	Sol_Status	Solution Status Code	I	Aux	(NA)	1: Optimal 2: Feasible 3: Infeasible 4: Error optimizing 5: Error reading data 6: Error initializing

Table "Year"

Key	Field	Description	Type	Origin	Model	Remarks
k	y_code	Code of Period (year)	A12	Cal (I)	NA	Y, set of periods, from G.Year_Ini thr G.Year_End
k(H)	y	Index of Period (year)	I	Cal (H)	y	Y, set of periods
	oscn	Fixed costs for ships (1)	R	Dat	$oscn_y$	
	ocscn	Fixed costs for ships (2)	R	Dat	$ocscn_y$	
	oapn	Fixed costs for aircraft (1)	R	Dat	$oapn_y$	
	ocapn	Fixed costs for aircraft (2)	R	Dat	$ocapn_y$	
	oom	Fixed costs for O&M	R	Dat	oom_y	
	toa_up	Upper bound on budget	R	Dat	$\overline{toa}_y$	
	toa_lo	Lower bound on budget	R	Dat	$\underline{toa}_y$	
	Cumtoa_up	Upper bound on cumulative budget	R	Dat	$\overline{ctoa}_y$	
	Cumtoa_lo	Lower bound on cumulative budget	R	Dat	$\underline{ctoa}_y$	
	max_ssab	Maximum set aside budget for ships	R	Dat	$\overline{ssab}_y$	
	max_asab	Maximum set aside budget for aircraft	R	Dat	$\overline{asab}_y$	
	Alpha_BPlus	Penalty for expenses excess	R	Dat	$bpen_y^+$	
	Alpha_BMinus	Penalty for expenses deficit	R	Dat	$bpen_y^-$	
	Alpha_CumBPlus	Penalty for cumulative expenses excess	R	Dat	$cbpen_y^+$	
	Alpha_CumBMinus	Penalty for cumulative expenses deficit	R	Dat	$cbpen_y^-$	
	CSBudget	Committed budget due to ship production on the way	R	Cal (H)	$csbudget_y$	
	SBudget_y	Required Ship Budget (before incremental rate)	R	Res	-	Update when SPROC(s,p,y) changes
	SBudget	Required Ship Budget	R	Res	$SBudget_y$	Update when SPROC(s,p,y) changes
	SSABudget	Set Aside Budget for Ships	R	Res	$SSABudget_y$	Update when SALabor(p,y) changes
	Abudget_y	Required Aircraft Budget (before incremental rate)	R	Res	-	Update when APROC(a,y) changes (after ASEG(a,y) updated)
	Abudget	Required Aircraft Budget	R	Res	$ABudget_y$	Update when APROC(a,y) changes (after ASEG(a,y) updated)
	ASABudget	Set Aside Budget for Aircraft	R	Res	$ASABudget_y$	
	OMSBudget_y	Required O&M Budget for ships	R	Res	"Cal"	Update when OMShip(s,y) changes
	OMABudget_y	Required O&M Budget for aircraft	R	Res	"Cal"	Update when AINV(a,y), changes

Key	Field	Description	Type	Origin	Model	Remarks
	OMBudget_y	Required O&M Budget (before incremental rate)	R	Res	-	Update when OMSBudget_y(y), OMABudget_y(y) change
	OMBudget	Required O&M Budget	R	Res	$OMBudget_y$	Update when OMSBudget_y(y), OMABudget_y(y) change
	Budget	Required Budget	R	Res	$Budget_y$	Update when SBudget(y), ABudget(y), OMBudget(y), SSABudget(y), ASABudget(y), change
	CumBudget	Required Cumulative Budget	R .	Res	"Cal"	Update when Budget(y) changes
	F_BPlus_y	Expenses excess penalty	R	Res	"Cal"	Update when Budget(y) changes
	F_CumBPlus_y	Cumulative expenses excess penalty	R	Res	"Cal"	Update when CumBudget(y) changes
	F_BMinus_y	Expenses deficit penalty	R	Res	"Cal"	Update when Budget(y) changes
	F_CumBMinus_y	Cumulative expenses deficit penalty	R	Res	"Cal"	Update when CumBudget(y) changes
	F_LPlus_y	Labor excess penalty	R	Res	"Cal"	Update when F_LPlus_py(p,y) changes
	F_LMinus_y	Labor deficit penalty	R	Res	"Cal"	Update when F_LMinus_py(p,y) changes
	F_B_y	Budget penalty	R	Res	"Cal"	Update when F_BPlus_y(y), F_BMinusy(y) change
	F_CumB_y	Cumulative budget penalty	R	Res	"Cal"	Update when F_CumBPlus_y(y), F_CumBMinusy(y) change
	F_L_y	Labor penalty	R	Res	"Cal"	Update when F_LPlus_y(y), F_LMinusy(y) change
	F_SM_y	Ship-Mission penalty	R	Res	"Cal"	Update when F_SM_smy (sm,y) changes
	F_AM_y	Air-Mission penalty	R	Res	"Cal"	Update when F_AM_amy (am,y) changes
	F_y	Total penalty in the year	R	Res	"Cal"	Update when F_B_y(y), F_L_y(y), F_SM_y(y), S_AM_y(y), change

Table "Ship"

Key	Field	Description	Type	Origin	Model	Remarks
k	s_code	Code of Ship class	A12	Dat	s	S, set of ship classes
k(H)	s	Index of Ship class	I	Cal (H)	(NA)	S, set of ship classes
	SInv_0	Initial number of ships	I	Dat	$sinv_s^0$	
	Max_SInv	Maximum ship inventory	I	Dat	$\overline{sinv_s}$	

Table "Aircraft"

Key	Field	Description	Type	Origin	Model	Remarks
k	a_code	Code of Aircraft type	A12	Dat	a	A, set of aircraft
k(H)	a	Index of Aircraft type	I	Cal (H)	(NA)	S, set of aircraft
	AInv_0	Initial number of aircraft	I	Dat	$ainv_a^0$	
	Max_AInv	Maximum aircraft inventory	I	Dat	$\overline{ainv_a}$	
	Max_ATot	Maximum aircraft procured in the planning time	I	Dat	$\overline{atot_a}$	
	squad_size	Group size for aircraft procurement	I	Dat	(Not modeled in the formulation)	Solution is a multiple of *squad_size*
	n_seg	Number of segments	I	Dat	$|I_a|$	Same for all years
	Aby_before	Budgeting years before delivery for aircraft	I	Dat	ABb_a	Aircraft is paid at once in that year

Table "Plant"

Key	Field	Description	Type	Origin	Model	Remarks
k	p_code	Code of Plant	A12	Dat	p	P, set of plants
k(H)	p	Index of Plant	I	Cal(H)	(NA)	P, set of plants
	lcrate	Labor cost rate of reference for setting aside labor and budget	R	Dat	$lcrate$	
	Alpha_LPlus	Penalty for labor excess	R	Dat	$lpen_p^+$	
	Alpha_LMinus	Penalty for labor deficit	R	Dat	$lpen_p^-$	

Table "Ship-Mission"

Key	Field	Description	Type	Origin	Model	Remarks
k	sm_code	Code of Ship-Mission	A12	Dat	$m \in M^S$	
k(H)	sm	Index of Ship-Mission	I	Cal(H)	(NA)	
	Alpha_SM	Penalty for failing to complete Ship-Mission	R	Dat	$smpen_m$	

Table "Air-Mission"

Key	Field	Description	Type	Origin	Model	Remarks
k	am_code	Code of Air-Mission	A12	Dat	$m \in M^A$	
k(H)	am	Index of Air-Mission	I	Cal(H)	(NA)	
	Alpha_AM	Penalty for failing to complete Air-Mission	R	Dat	$ampen_m$	

Table "Ship-Year"

Key	Field	Description	Type	Origin	Model	Remarks
k	s_code	Code of Ship class	A12	Dat	s	
k(H)	s	Index of Ship class	I	Cal (H)	(NA)	
k	y_code	Code of Period (year)	A12	Cal (I)	NA	
k(H)	y	Index of Period (year)	I	Cal (H)	y	
	CSInv	Committed inventory of ships for the year due to production in progress	I	Cal (H)	$csproc_{sy}$	SUM of SPY.CSInv_spy over plants
	oldS_cum_min	Cumulative ships to retire (minimum)	I	Dat	$\underline{csret}_{sy}$	
	oldS_cum_max	Cumulative ships to retire (maximum)	I	Dat	$\overline{csret}_{sy}$	
	oldS_min	Individual ships to retire (minimum)	I	Dat	$\underline{sret}_{sy}$	
	oldS_max	Individual ships to retire (maximum)	I	Dat	$\overline{sret}_{sy}$	
	OMShip	O&M costs for ships	R	Dat	$omship_{sy}$	
	I_SRET	Initial solution for the ships retirement	I	F	(NA)	0 for the interface
	SPROC_sy	Number of ships delivered from all plants (including committed)	I	Res	(NA)	
	SRET	Number of ships retired	I	Res	$SRet_{sy}$	Main Decision Variable
	SINV	Inventory of ships	I	Res	$SInv_{sy}$	Update when SPROC(s,p,y), SRET(s,y) change
	SBudget_sy	Ship budget required for ships in the year (before incremental rate)	R	Res	(NA)	Update when SPROC(s,p,y), changes
	OMSBudget_sy	O&M budget required for ships in the year	R	Res	(NA)	Update when SINV (s,y) changes

Table "Ship-Plant"

Key	Field	Description	Type	Origin	Model	Remarks
k	s_code	Code of Ship class	A12	Dat	s	
k(H)	s	Index of Ship class	I	Cal (H)	(NA)	
k	p_code	Code of Plant	A12	Dat	p	
k(H)	p	Index of Plant	I	Cal(H)	(NA)	
	Allowed_sp	Whether we can produce new ships at the plant or not. If not, the Ship-Plant pair is used only in the interface to calculate committed labor, committed inventory, etc.	L	Dat (I)/ Cal (H)	P_s	This field is not exported. Only those records with Allowed_sp= 'Yes' are exported.
	Sby_before	Budgeting years before delivery	I	Dat	SBb_{sp}	
	Sby_after	Budgeting years after delivery	I	Dat	SBa_{sp}	

Key	Field	Description	Type	Origin	Model	Remarks
	Scy_before	Construction years before delivery	I	Dat	SCb_{sp}	
	Scy_after	Budgeting years after delivery	I	Dat	SCa_{sp}	
	Max_STot	Maximum total number of ships class s procured from plant p	I	Dat	$stot_{sp}$	0 if Allowed_sp= 'No'
	relation	Index of parallel delivery (e.g., for constraint (16))	I	Dat	*(Not in the model formulation yet)*	0 if none; same index implies relationship

Table "Ship-Ship-Mission"

Key	Field	Description	Type	Origin	Model	Remarks
k	s_code	Code of Ship class	A12	Dat	s	
k(H)	s	Index of Ship class	I	Cal (H)	(NA)	
k	sm_code	Code of Ship-Mission	A12	Dat	$m \in M^s$	
k(H)	sm	Index of Ship-Mission	I	Cal(H)	(NA)	
	Allowed_ssm (*)	Whether a ship class can perform a Ship-Mission or not	L	Cal (H)	S_m	'Yes' if the record exists
	SEff	Effectiveness rating	R	Dat	$seff_{sm}$	If =0, the record can be deleted

(*) The field may be omitted in the database assuming that only those existing records correspond to Allowed_ssm= 'Yes'

Table "Aircraft-Year"

Key	Field	Description	Type	Origin	Model	Remarks
k	a_code	Code of Aircraft type	A12	Dat	a	
k(H)	a	Index of Aircraft type	I	Cal (H)	(NA)	
k	y_code	Code of Period (year)	A12	Cal (I)	(NA)	
k(H)	y	Index of Period (year)	I	Cal(H)	y	
	qamin	Minimum number of aircraft to be procured	I	Dat	$\underline{aproc}_{ay}$	
	qamax	Maximum number of aircraft to be procured	I	Dat	$\overline{aproc}_{ay}$	
	CAInv	Committed procurement of aircraft due to production in progress	I	Dat	$caproc_{ay}$	
	oldA_cum_min	Cumulative aircraft to retire (minimum)	I	Dat	$\underline{caret}_{ay}$	
	oldA_cum_max	Cumulative aircraft to retire (maximum)	I	Dat	$\overline{caret}_{ay}$	
	oldA_min	Individual aircraft to retire (minimum)	I	Dat	$\underline{aret}_{ay}$	
	oldA_max	Individual aircraft to retire (maximum)	I	Dat	$\overline{aret}_{ay}$	
	OMAir	O&M cost for aircraft	R	Dat	$omair_a$	
	I_APROC	Initial solution for the aircraft procurement	I	F	(NA)	0 for the interface
	I_ARET	Initial solution for the aircraft retirement	I	F	(NA)	0 for the interface
	min_ASEG	Minimum segment with a feasible procurement	I	Cal (H)	(NA)	Relative to squad_size
	min_APROC	Minimum feasible procurement	I	Cal (H)	(NA)	Relative to squad_size
	APROC	Number of aircraft delivered	I	Res	*"Cal"*	New: $APROC_{ay}$

Key	Field	Description	Type	Origin	Model	Remarks
					$(\sum_{i \in I_a} AProc_{ayi})$	Main Decision Variable
	APROC	Number of aircraft delivered (including committed)	I	Res	(NA)	
	ARET	Number of aircraft retired	I	Res	$ARet_{ay}$	Main Decision Variable
	AINV	Inventory of aircraft	I	Res	$AInv_{ay}$	Update when APROC(a,y), ARET(a,y) change
	ABudget_ay	Air budget required for aircraft in the year (before incremental rate)	R	Res	(NA)	Update when APROC(a,y), changes

Table "Aircraft-Air-Mission"

Key	Field	Description	Type	Origin	Model	Remarks
k	a_code	Code of Aircraft type	A12	Dat	a	$a \in a_m$
k(H)	a	Index of Aircraft type	I	Cal (H)	(NA)	$a \in a_m$
k	am_code	Code of Air-Mission	A12	Dat	$m \in M^A$	
k(H)	am	Index of Air-Mission	I	Cal(H)	(NA)	
	Allowed_aam (*)	Whether an aircraft type can perform an Air-Mission or not	L	Cal (H)	A_m	'Yes' if the record exists
	AEff	Effectiveness rating	R	Dat	$aeff_{am}$	If =0, the record can be deleted

(*) The field may be omitted in the database assuming that only those existing records correspond to Allowed_aam = 'Yes'

Table "Plant-Year"

Key	Field	Description	Type	Origin	Model	Remarks
k	p_code	Code of Plant	A12	Dat	p	
k(H)	p	Index of Plant	I	Cal(H)	(NA)	
k	y_code	Code of Period (year)	A12	Cal (I)	(NA)	
k(H)	y	Index of Period (year)	I	Cal (H)	y	
	max_sal	Maximum labor set aside	I	Dat	$\overline{sal}_{py}$	
	CLabor	Committed labor due to production in progress	I	Cal (H)	$clabor_{py}$	Relative to CSInv
	pcap_up	Maximum labor	I	Dat	$\overline{pcap}_{py}$	
	pcap_lo	Minimum labor	I	Dat	$\underline{pcap}_{py}$	
	SALabor	Labor set aside	I	Res	$SALabor_{py}$	Determines $SSAB_y$
	LABOR	Required Labor	I	Res	$Labor_{py}$	Update when SPROC(s,p,y), SALabor(y) changes
	F_LPlus_py	Penalty for labor excess	R	Res	"Cal"	Update when LABOR(p,y) changes
	F_LMinus_py	Penalty for labor deficit	R	Res	"Cal"	Update when LABOR(p,y) changes
	F_L_py	Labor penalty	R	Res	"Cal"	Update when F_LPlus_py(p,y), F_LMinus_py(p,y) change (NOT needed later, though)

Table "Ship-Mission-Year"

Key	Field	Description	Type	Origin	Model	Remarks
k	sm_code	Code of Ship-Mission	A12	Dat	$m \in M^S$	
k(H)	sm	Index of Ship-Mission	I	Cal(H)	(NA)	
k	y_code	Code of Period (year)	A12	Cal (I)	(NA)	
k(H)	y	Index of Period (year)	I	Cal(H)	y	
	smreq	Number of Ship-Missions required	I	Dat	$smreq_{my}$	
	SMInv	Number ships that can perform a Ship-Mission	I	Res	$SMInv_{my}$	Update when SInv(s,y) changes
	SMEff	Overall effectiveness for a Ship-Mission	R	Res	$SMEff_{my}$	Update when SInv(s,y) changes
	F_SM_smy	Penalty for Ship-Mission shortfall	R	Res	"Cal"	Update when SMEff(sm,y) changes

Table "Air-Mission-Year"

Key	Field	Description	Type	Origin	Model	Remarks
k	am_code	Code of Air-Mission	A12	Dat	$m \in M^A$	
k(H)	am	Index of Air-Mission	I	Cal(H)	(NA)	
k	y_code	Code of Period (year)	A12	Cal (I)	(NA)	
k(H)	y	Index of Period (year)	I	Cal(H)	y	
	amreq	Number of Air-Missions required	I	Dat	$amreq_{my}$	
	AMInv	Number aircraft that can perform an Air-Mission	I	Res	$AMInv_{my}$	Update when AInv(a,y) changes
	AMEff	Overall effectiveness for an Air-Mission	R	Res	$AMEff_{my}$	Update when AInv(a,y) changes
	F_AM_amy	Penalty for Air-Mission shortfall	R	Res	"Cal"	Update when AMEff(am,y) changes

Table "Ship-Plant-Year"

Key	Field	Description	Type	Origin	Model	Remarks
k	s_code	Code of Ship class	A12	Dat	s	
k(H)	s	Index of Ship class	I	Cal (H)	(NA)	
k	p_code	Code of Plant	A12	Dat	p	
k(H)	p	Index of Plant	I	Cal(H)	(NA)	
k	y_code	Code of Period (year)	A12	Cal (I)	(NA)	
k(H)	y	Index of Period (year)	I	Cal(H)	y	
	qsmin	Minimum number of ships to be procured	I	Dat	$\underline{sproc}_{spy}$	
	qsmax	Maximum number of ships to be procured	I	Dat	$\overline{sproc}_{spy}$	
	CSInv_spy	Committed number of ships due to production in progress	I	Dat	(NA)	Update SP.*CSInv*
	I_SPROC	Initial solution for the ship procurement	I	F	(NA)	0 for the interface
	SPROC	Number of ships delivered	I	Res	$SPROC_{syp}$	New: $SPROC_{spy}$ Main Decision Variable

Table "Aircraft-Year-Segment"

Key	Field	Description	Type	Origin	Model	Remarks
k	a_code	Code of Aircraft type	A12	Dat	a	
k(H)	a	Index of Aircraft type	I	Cal (H)	(NA)	
k	y_code	Code of Period (year)	A12	Cal (I)	(NA)	
k(H)	y	Index of Period (year)	I	Cal(H)	y	
k	i	Index of segment	I	Cal (I)	$i \in I_a$	$i=1...n\_seg$ in 'Aircraft'
	inc_lo	Minimum number of aircraft in the segment	I	Dat	$\underline{inc}_{ayi}$	
	inc_up	Maximum number of aircraft in the segment	I	Dat	$\overline{inc}_{ayi}$	
	aacost	Lineal cost in the segment	R	Dat	$aacost_{ayi}$	
	abcost	Independent term of cost in the segment	R	Dat	$abcost_{ayi}$	
	ASEG	Whether the purchase is in the segment or not	L	Res	AP_{ayi}	Update when APROC(a,y) changes

Table "Ship-Plant-Quantity-Budgeting Year Before"

Key	Field	Description	Type	Origin	Model	Remarks
k	s_code	Code of Ship class	A12	Dat	s	Only ships that can be produced
k(H)	s	Index of Ship class	I	Cal (H)	(NA)	
k	p_code	Code of Plant	A12	Dat	p	Only plants that may produce the ship
k(H)	p	Index of Plant	I	Cal(H)	(NA)	
k	q	Index of Number Ships	I	Cal (I)	q	For $q=1...qmax$ in 'Ship'
k	l	Index of budgeting year (before delivery), i.e., n=0 means delivery year, n=1 year before, ...	I	Cal (I)	l	For l=0...Sby_before-1 in 'Ship-Plant' (*)
	scost_before	Ship cost (installment)	R	Dat	$scostb_{spql}$	

(*) In the heuristic array structures the indices l=0, ...,Sby_before are stored as l=1,...,Sby_before+1, respectively

Table "Ship-Plant-Quantity-Budgeting Year After"

Key	Field	Description	Type	Origin	Model	Remarks
k	s_code	Code of Ship class	A12	Dat	s	Only ships that can be produced
k(H)	s	Index of Ship class	I	Cal (H)	(NA)	
k	p_code	Code of Plant	A12	Dat	p	Only plants that may produce the ship
k(H)	p	Index of Plant	I	Cal (H)	(NA)	
k	q	Index of Number Ships	I	Cal (I)	q	For $q=1...qmax$ in 'Ship'
k	ll	Index of budgeting year (after delivery), i.e., n=0 means delivery year, n=1 year before, ...	I	Cal (I)	l	For ll=1...Sby_after in 'Ship-Plant'
	scost_after	Ship cost (installment)	R	Dat	$scosta_{spql}$	

Table "Ship-Plant-Quantity-Construction Year Before"

Key	Field	Description	Type	Origin	Model	Remarks
k	s_code	Code of Ship class	A12	Dat	s ·	Only ships that can be produced
k(H)	s	Index of Ship class	I	Cal (H)	(NA)	
k	p_code	Code of Plant	A12	Dat	p	Only plants that may produce the ship
k(H)	p	Index of Plant	I	Cal(H)	(NA)	
k	q	Index of Number Ships	I	Cal (I)	q	For $q=1...qmax$ in 'Ship'
k	n	Index of construction year (before delivery), i.e., n=0 means delivery year, n=1 year before, ...	I	Cal (I)	n	For $n=0...$Scy_before-1 in 'Ship-Plant' (*)
	sw_before	Number workers needed	I	Dat	$sworkb_{spqn}$	

(*) In the heuristic array structures the indices n=0, ...,Scy_before are stored as n=1,...,Scy_before+1, respectively

Table "Ship-Plant-Quantity-Construction Year After"

Key	Field	Description	Type	Origin	Model	Remarks
k	s_code	Code of Ship class	A12	Dat	s	Only ships that can be produced
k(H)	s	Index of Ship class	I	Cal (H)	(NA)	
k	p_code	Code of Plant	A12	Dat	p	Only plants that may produce the ship
k(H)	p	Index of Plant	I	Cal(H)	(NA)	
k	q	Index of Number Ships	I	Cal (I)	q	For $q=1...qmax$ in 'Ship'
k	nn	Index of construction year (after delivery), i.e., n=1 year after, ...	I	Cal (I)	n	For $nn=1...$Scy_after in 'Ship-Plant'
	sw_after	Number workers needed	I	Dat	$sworka_{spqn}$	

Table "Control"

Key	Field	Description	Type	Origin	Model	Remarks
	MR_wF_SM	Weight for F_SM in Mission Rank formula	R	Ctr		
	MR_wF_AM	Weight for F_AM in Mission Rank formula	R	Ctr		
	SMR_wF_SM	Weight for F_SM in Ship-Mission Rank formula	R	Ctr		
	SMR_wF_L	Weight for F_L in Ship-Mission Rank formula	R	Ctr		
	SMR_wF_B	Weight for F_B in Ship-Mission Rank formula	R	Ctr		
	SMR_IF_F_Impr	Request for Improvement in F to accept a candidate in Ship-Mission Rank	L	Ctr		
	SMR_IF_F_SM_Impr	Request for Improvement in F_SM to accept a candidate in Ship-Mission Rank	L	Ctr		
	AMR_wF_AM	Weight for F_AM in Air-Mission Rank formula	R	Ctr		
	AMR_wF_B	Weight for F_B in Air-Mission Rank formula	R	Ctr		
	AMR_IF_F_Impr	Request for Improvement in F to accept a candidate in Air-Mission Rank	L	Ctr		
	AMR_IF_F_AM_Impr	Request for Improvement in F_AM to accept a candidate in Air-Mission Rank	L	Ctr		
	LR_wF_SM	Weight for F_SM in Labor Rank formula	R	Ctr		
	LR_wF_L	Weight for F_L in Labor Rank formula	R	Ctr		
	LR_wF_B	Weight for F_B in Labor Rank formula	R	Ctr		
	LR_IF_F_Impr	Request for Improvement in F to accept a candidate in Labor Rank	L	Ctr		
	LR_IF_F_L_Impr	Request for Improvement in F_L to accept a candidate in Labor Rank	L	Ctr		
	SRR_wIncF_SM	Weight for ΔF_SM in Ship Retirement Rank formula	R	Ctr		
	SRR_wIncOMS	Weight for ΔOM Ships in Ship Retirement Rank formula	R	Ctr		
	SRR_Max_Diff	Maximum difference between ΔF_SM and ΔOM Ships to consider a candidate in Ship Retirement Rank formula	R	Ctr		
	ARR_wIncF_AM	Weight for ΔF_AM in Aircraft Retirement Rank formula	R	Ctr		
	ARR_wIncOMA	Weight for ΔOM Air in Aircraft Retirement Rank formula	R	Ctr		
	ARR_Max_Diff	Maximum difference between ΔF_AM and ΔOM Air to consider a candidate in Aircraft Retirement Rank formula	R	Ctr		

A.3   Data and Result Files

The following is an outline of the processes involved in the use of the CIPA system:

a) The User Creates a Case Using the System Interface-Database.
b) The Case is Exported for Optimization: ASCII Data Files.
c) The Data Files are Read and the Case is Optimized.
d) The Results are Exported: ASCII Data Files.
e) The Interface Reads the Result Files for the Case.
f) The User Consults the Results.
g) The User Saves the Case and/or Modifies the Data.

The steps (b) and (d) above refer to processes that require communication between the user interface and the algorithm. In order to integrate these two subsystems ASCII data files will be created. In the first part of this section we refer to the data flows from the interface to the algorithm. The second part explains the files produced by the algorithm containing results to be used in the interface.

A.2.1 Interface to Algorithm Data files: Case Data

What data fields need to be exported?

Origin	Export
Dat	Yes
Dat (I)	No
Cal, Cal(I)	Yes
Cal(H)	No
F	Yes
Ctr	Yes
Res, Aux	No

Table. Data I/O.

Data formats are standardized as follows:

- Integer data: 12 digits (I12).
- Real data: 12 digits distributed as follows: two decimal digits, one digit for the point, one digit for the minus sign (if any), and eight or nine digits for the integer part (F12.2).
- Boolean/logical data: Will be treated as integer data, that is, 1 for "Yes" and 0 for "No," exported as 12-digit integers.
- Alphanumeric: Except for the "Plan_Code" field in table "General," all the other alphanumeric data are codes with 12 characters.

Indexed Data Files

All the data files associated with tables containing indices (i.e., all but "General" and "Control") have the following similar structure:

97

- File names: The location of all the data files will be the < *path* \Data> folder. *path* is the location of the program CIPA.exe (heuristic algorithm). The name of the files is provided by the indices of the table grouped together with the extension ".dat" as follows:

Table Real Name	File Name
Year	Y.dat
Ship	S.dat
Aircraft	A.dat
Plant	P.dat
Ship-Mission	SM.dat
Air mission	AM.dat
Ship – Year	SY.dat
Ship – Plant	SP.dat
Ship – Ship-Mission	SSM.dat
Aircraft – Year	AY.dat
Aircraft – Air-Mission	AAM.dat
Plant – Year	PY.dat
Ship-Mission – Year	SMY.out
Air-Mission – Year	AMY.out
Ship – Plant – Year	SPY.dat
Aircraft – Year – Segment	AYI.dat
Ship – Plant – Quantity – Budgeting year before	SPQL.dat
Ship – Plant – Quantity – Budgeting year after	SPQLL.dat
Ship – Plant – Quantity – Construction year before	SPQN.dat
Ship – Plant – Quantity – Construction year after	SPQNN.dat

Table. Tables and data files.

- File structure and contents:

  ➤ Line 1 is used for comments (e.g., headers with field names). It may be left blank.
  ➤ From line 2 to the end of the file there is one record per line. There is a fixed format as specified below.
  ➤ Every field will be associated a width of 12 columns and there will be three blank spaces between fields. Therefore:

    ➔ The first field starts in column 1 and ends in column 12.
    ➔ The second field starts in column 16 and ends in column 27.
    ➔ The third field starts in column 31 and ends in column 42.
    ➔ And so forth (46-57, 61-72, 76-87, ...).

Examples:

File Y.dat

y_code	oscn	ocscn	oapn	ocapn	oom	...
FY06	0.00	532.71	0.00	4356.30	4839.92	
FY07	35.00	634.11	0.00	4874.09	4774.40	
FY08	0.00	282.25	0.00	5323.42	4765.31	
FY09	35.00	516.20	0.00	4721.60	4661.75	
FY10	0.00	1660.92	0.00	5509.91	4669.37	
FY11	35.00	391.85	0.00	6101.37	4537.59	
...						

The fields are Y.y_code, Y.oscn, Y.ocscn, Y.oapn, Y.ocapn, Y.oom, Y.toa_up, Y.toa_lo, Y.Cumtoa_up, Y.Cumtoa_lo, Y.max_ssab, Y.max_asab, Y.Alpha_BPlus, Y.Alpha_BMinus, Y.Alpha_CumBPlus, Y.Alpha_CumBMinus

File S.dat

s_code	SInv_0	Max_SInv
DDG	46	999999999
DD21	0	999999999
CVX	0	999999999
SSN774	2	999999999
LHX	0	999999999
FFG	24	999999999
DD	19	999999999
CG	27	999999999
SSN688	45	999999999
SSN21	3	999999999
CVN68	9	999999999

The fields are S.s_code, S.SInv_0, S.Max_SInv

File A.dat

a_code	AInv_0	Max_AInv	Max_ATot	squad_size	n_seg	...
JSFN	0	999999999	999999999	12	4	
F18EF	218	999999999	999999999	4	4	
F18AB	184	999999999	999999999	4	4	
F18CD	467	999999999	999999999	4	4	
F14	74	999999999	999999999	4	4	

The fields are A.a_code, A.AInv_0, A.Max_AInv, A.Max_ATot, A.squad_size, A.n_seg, A.Aby_before

File P.dat

p_code	lcrate	Alpha_LPlus	Alpha_LMinus
Bath	0.58	0.45	0.60
Ingals	0.60	0.22	0.29
News	0.60	0.45	0.61
Eboat	0.30	0.48	0.64
Avon	0.10	0.41	0.55

The fields are P.p_code, P.lcrate, P.Alpha_LPlus, P.Alpha_LMinus

File SM.dat

sm_code	Alpha_SM
combatant	1381.14
combatantCG	1189.35
carrier	3923.66
attack	1638.68
amphibH	2117.05
amphibS	478.37
amphibP	780.32

The fields are SM.sm_code, SM.Alpha_SM

File AM.dat

am_code	Alpha_AM
fighter	78.51

The fields are AM.am_code, AM.Alpha_AM

File SY.dat

s_code	y_code	oldS_cum_min	oldS_cum_max	oldS_min	oldS_max ...
DDG	FY06	0	999999	0	1
DDG	FY07	0	999999	0	2
DDG	FY08	0	999999	0	5
DDG	FY09	0	999999	0	5
...					
DDG	FY25	0	999999	0	23
DD21	FY06	0	999999	0	10
DD21	FY07	0	999999	0	11
...					
LPD17	FY24	0	999999	0	20
LPD17	FY25	0	999999	0	25

The fields are SY.s_code, SY.y_code, SY.oldS_cum_min, SY.oldS_cum_max, SY.oldS_min, SY.oldS_max, SY.OMShip, SY.I_SRET

File SP.dat

s_code	p_code	Sby_before	Sby_after	Scy_before	Scy_after ...
DDG	Bath	5	0	5	0
DDG	Ingals	4	0	4	0
DD21	Bath	4	0	4	0
DD21	Ingals	4	0	4	0
CVX	News	9	0	7	0
SSN774	News	8	0	6	0
SSN774	Eboat	8	0	6	0
LHX	Ingals	6	0	6	0

The fields are SP.s_code, SP.p_code, SP.Sby_before, SP.Sby_after, SP.Scy_before, SP.Scy_after, SP.Max_STot, SP.relation

File SSM.dat

s_code	sm_code	SEff
DDG	combatant	1.00
DD21	combatant	1.00
CVX	carrier	1.00
SSN774	attack	1.00
LHX	amphibH	1.00
FFG	combatant	1.00
DD	combatant	1.00
...		

The fields are SSM.s_code, SSM.sm_code, SSM.SEff

100

File AY.dat

a_code	y_code	qamin	qamax	CAInv	oldA_cum_min..
JSFN	FY06	0	0	0	0
JSFN	FY07	0	0	0	0
JSFN	FY08	0	0	0	0
JSFN ·	FY09	0	0	0	0
JSFN	FY10	0	0	0	0
JSFN	FY11	0	55	0	0
JSFN	FY12	0	55	0	0
...					
JSFN	FY25	0	55	0	0
F18EF	FY06	0	55	48	0
F18EF	FY07	0	55	48	0
...					
F14	FY25	0	0	0	74

The fields are AY.a_code, AY.y_code, AY.qamin, AY.qamax, AY.CAInv, AY.oldA_cum_min, AY.oldA_cum_max, AY.oldA_min, AY.oldA_max, AY.OMAir, AY.I_APROC, AY.I_ARET

File AAM.dat

a_code	am_code	AEffect
JSFN	fighter	1.00
F18EF	fighter	1.00
F18AB	fighter	1.00
F18CD	fighter	1.00
F14	fighter	1.00

The fields are AAM.a_code, AAM.am_code, AAM.AEff

File PY.dat

p_code	y_code	max_sal	pcap_up	pcap_lo
Bath	FY06	5000	9000	2800
Bath	FY07	5000	9000	2800
Bath	FY08	5000	9000	2800
Bath	FY09	5000	9000	2800
...				
Bath	FY25	5000	9000	2800
Ingals	FY06	5000	17000	6500
Ingals	FY07	5000	17000	6500
Ingals	FY08	5000	17000	6500
...				

The fields are PY.p_code, PY.y_code, PY.max_sal, PY.pcap_up, PY.pcap_lo

File SMY.dat

sm_code	y_code	smreq
combatant	FY06	98
combatant	FY07	98
combatant	FY08	98
combatant	FY09	98
combatant	FY10	98
...		

The fields are SMY.sm_code, SMY.y_code, SMY.smreq

File AMY.dat

am_code	y_code	amreq
fighter	FY06	895
fighter	FY07	895
fighter	FY08	895
fighter	FY09	895
fighter	FY10	895
fighter	FY11	895
fighter	FY12	895
...		

The fields are AMY.am_code, AMY.y_code, AMY.amreq

File SPY.dat

s_code	p_code	y_code	qsmin	qsmax	CSInv_spy	...
DDG	Bath	FY06	0	2	0	
DDG	Bath	FY07	0	2	0	
DDG	Bath	FY08	0	2	0	
DDG	Bath	FY09	0	2	0	
DDG	Bath	FY10	0	2	0	
DDG	Bath	FY11	0	2	2	
DDG	Bath	FY12	0	2	2	
...						
DDG	Ingals	FY25	0	0	0	
DD21	Bath	FY06	0	2	0	
DD21	Bath	FY07	0	2	0	
DD21	Bath	FY08	0	2	0	
DD21	Bath	FY09	0	2	0	
...						

The fields are SPY.s_code, SPY.p_code, SPY.y_code, SPY.qsmin, SPY.qsmax, SPY.CSInv_spy, SPY.I_SPROC

File AYI.dat

a_code	y_code	i	inc_lo	inc_up	aacost
JSFN	FY06	1	0	0	0.00
JSFN	FY06	2	24	30	49.09
JSFN	FY06	3	30	40	47.04
JSFN	FY06	4	40	55	45.51
JSFN	FY07	1	0	0	0.00
JSFN	FY07	2	24	30	49.09
JSFN	FY07	3	30	40	47.04
JSFN	FY07	4	40	55	45.51
JSFN	FY08	1	0	0	0.00
...					
JSFN	FY25	4	40	55	45.51
F18EF	FY06	1	0	0	0.00
F18EF	FY06	2	24	30	45.27
F18EF	FY06	3	30	40	39.91
F18EF	FY06	4	40	55	36.30
F18EF	FY07	1	0	0	0.00
F18EF	FY07	2	24	30	45.27
F18EF	FY07	3	30	40	39.91
...					

The fields are AYI.a_code, AYI.y_code, AYI.i, AYI.inc_lo, AYI.inc_up, AYI.aacost, AYI.abcost

File SPQL.dat

s_code	p_code	q	l		scost_before
DDG	Bath	1		5	1405.72
DDG	Bath	2		5	2212.16
DDG	Ingals	1		4	1405.72
DDG	Ingals	2		4	2212.16
DD21	Bath	1		4	1822.06
DD21	Bath	2		4	2394.15
DD21	Ingals	1		4	1822.06
DD21	Ingals	2		4	2394.15
CVX	News	1		7	188.54
CVX	News	1		8	0.00
...					

The fields are SPQL.s_code, SPQL.p_code, SPQL.q, SPQL.l, SPQL.scost_before

1. **Remark:** If any of these records are omitted, it is assumed a value of scost_before=0.0. For instance, in the example, the purchase of $q=1$ ship class s_code=DDG in p_code=Ingals requires only one payment to be made $l=4$ years before delivery (in the amount of $1,405.72). Since no other payments are indicated, we assume that the remaining payments from $l=3$ through $l=0$ years before delivery are all equal to zero.
2. Notice that the index l runs from $l=0$ to $l=SP.Sby\_before -1$.

File SPQLL.dat

s_code	p_code	.q	ll	scost_after

The fields are SPQLL.s_code, SPQLL.p_code, SPQLL.q, SPQLL.ll, SPQLL.scost_after

1. See also "**Remark 1**" for SPQL.dat file.
2. Note that in this example SPQLL.dat has no records because there are no budgeting years after delivery for any ship (SP.Sby_after=0).
3. In general, the index ll runs from $ll=1$ to $ll=SP.Sby\_after$.

File SPQN.dat

s_code	p_code	q	n	sw_before
DDG	Bath	1	0	357
DDG	Bath	1	1	681
DDG	Bath	1	2	824
DDG	Bath	1	3	122
DDG	Bath	1	4	132
DDG	Bath	1	5	132
DDG	Bath	2	0	714
DDG	Bath	2	1	1362
DDG	Bath	2	2	1648
DDG	Bath	2	3	244
DDG	Bath	2	4	264
DDG	Bath	2	5	264
DDG	Ingals	1	0	558
DDG	Ingals	1	1	1037
...				

The fields are SPQN.s_code, SPQN.p_code, SPQN.q, SPQN.n, SPQN.sw_before

1. **Remark 1:** If any of these records are omitted, it is assumed a value of sw_before=0.
2. Notice that the index n runs from $n=0$ to $n=SP.Scy\_before -1$.

103

File SPQNN.dat

s_code	p_code	q	nn	sw_after

The fields are SPQNN.s_code, SPQNN.p_code, SPQNN.q, SPQNN.nn, SPQNN.sw_after

1. See also "**Remark 1**" for SPQN.dat file.
2. Note that in this example SPQNN.dat has no records because there are no construction years after delivery for any ship (SP.Scy_after=0).
3. In general, the index nn runs from nn=1 to nn=SP.Scy_after.

Non-Indexed Data Files

For the other two tables that do not contain indices, the associated files and their formats are as follows:

- Table General:

  ➢ The associated file name is G.dat.
  ➢ It will be located in the *<path* \Data> folder.
  ➢ The first row of the file is a comment line.
  ➢ The second row contains the General.Plan_Code field in columns 1 through 50.
  ➢ The third row is a comment line.
  ➢ The fourth row is structured as follows:

    ▪ Columns 1 thru 12: General.frac
    ▪ Columns 16 thru 27: General.apn5
    ▪ Columns 31 thru 42: General.Isol_User
    ▪ Columns 46 thru 57: General.Gams_opt

Example:

File G.dat

```
Plan_Code
example number 1: baseline case
frac          apn5          ISol_User      Gams_Opt
     0.03          0.34          0              0
```

The field in Row 2 is G.Plan_Code
The fields in Row 4 are G.frac, G.apn5, G.ISol_SPROC, G.ISol_User, G.Gams_opt

- Table Control:

  ➢ The associated file name is Control.dat
  ➢ It will be located in the *<path*\Data> folder
  ➢ For the moment we may assume that this file is fixed, so we can use it for all the instances, being careful not to delete it every time we delete the other .dat files before running a new case

A.2.2 Algorithm to Interface Data files: Case Results

What data fields need to be imported?

Origin	Import
Res, Aux	Yes
(others)	No

104

Data formats for the results are standardized as follows:

- Integer data: 12 digits (I12).
- Real data: 12 digits distributed as follows: two decimal digits, one digit for the point, one digit for the minus sign (if any), and eight or nine digits for the integer part.
- Boolean/logical data: Will be treated as integer data, that is, 1 for "Yes" and 0 for "No," imported as 12 digit integers.
- Alphanumeric: As described in the type column for each field.

It is important to point out that the file G.out, containing general and non-indexed results, will always exist. This file is described later in this document. Once the value of the field General.Prog_Status comes out (after reading G.out) we will be able to read the other output files (described below) if General.Prog_Status=1. However, if General.Prog_Status=2, then G.out will be the only output file.

Indexed Result Files

All the result files associated with tables containing indices (i.e., all but "General") have the following similar structure:

- File names: The location of all the result files will be the *<path*\Results> folder. Likewise the files containing the data, the name of the result files is provided by the indices of the table grouped together, plus the extension ".out":

Table Real Name	File Name
Year	Y.out
Ship-Year	SY.out
Aircraft-Year	AY.out
Plant-Year	PY.out
Ship-Mission-Year	SMY.ot
Air-Mission-Year	AMY.out
Ship-Plant-Year	SPY.out
Aircraft-Year-Segment	AYI.out

- File structure and contents:
  - ➢ Line 1 is used for comments (e.g., headers with field names). It may be left blank.
  - ➢ From line 2 to the end of the file there is one record per line. There is a fixed format as specified below.
  - ➢ Every field will be associated a width of 12 columns and there will be three blank spaces between fields. Therefore:
    - ➔ The first field starts in column 1 and ends in column 12.
    - ➔ The second field starts in column 16 and ends in column 27.
    - ➔ The third field starts in column 31 and ends in column 42.
    - ➔ And so forth (46-57, 61-72, 76-87, ...).
    - ➔ There are no alphanumeric data in the indexed tables, so all the numeric fields fit in the specified room.

105

Example:

File Y.out

y_code	SBudget_y	SBudget	SSABudget	ABudget_y	ABudget	ASABudget ...
FY06	12962.10	13899.65	0.00	8038.74	16591.83	0.00
FY07	9967.92	10955.09	0 00	8618.67	16735.75	0.00
FY08	10845.90	11462.00	0.00	9121.92	17284.30	0.00
FY09	8499.41	9321.08	0.00	11206.09	17427.19	0.00
...						

The fields are Y.y_code, Y.SBudget_y, Y.SBudget, Y.SSABudget, Y.ABudget_y, Y.ABudget, Y.ASABudget, Y.OMSBudget_y, Y.OMABudget_y, Y.OMBudget_y, Y.OMBudget, Y.Budget, Y.CumBudget, Y.F_BPlus_y, Y.F_CumBPlus_y, Y.F_BMinus_y, Y.F_CumBMinus_y, Y.F_LPlus_y, Y.F_LMinus_y, Y.F_B_y, Y.F_CumB_y, Y.F_L_y, Y.F_SM_y, Y.F_AM_y, Y.F_y

File SY.out

s_code	y_code	SPROC_sy	SRET	SINV	SBudget_sy
DDG	FY06	3	0	49	2212.16
DDG	FY07	1	0	50	0.00
DDG	FY08	4	0	54	0.00
DDG	FY09	1	0	55	0.00
DDG	FY10	0	0	55	0.00
...					

The fields are SY.s_code, SY.y_code, SY.SPROC_sy, SY.SRET, SY.SINV

File AY.out

a_code	y_code	APROC	APROC_ay	ARET	AINV ...
JSFN	FY06	0	0	0	0
JSFN	FY07	0	0	0	0
JSFN	FY08	0	0	0	0
JSFN	FY09	0	0	0	0
JSFN	FY10	0	0	0	0
...					

The fields are AY.a_code, AY.y_code, AY.APROC, AY.APROC_ay, AY.ARET, AY.AINV, AY.ABudget_ay

File PY.out

p_code	y_code	SALabor	LABOR	F_LPlus_py	F_LMinus_py	F_L_py
Bath	FY06	0.00	4209 . ,	0.00	0.00	0.00
Bath	FY07	0.00	6180	0.00	0.00	0.00
Bath	FY08	0.00	7292	0.00	0.00	0.00
Bath	FY09	0.00	8174	0.00	0.00	0.00
Bath	FY10	0.00	9292	131.40	0.00	131.40
Bath	FY11	0.00	8358	0.00	0.00	0.00
...						

The fields are PY.p_code, PY.y_code, PY.SALabor, PY.LABOR, PY.F_LPlus_py, PY.F_LMinus_py, PY.F_L_py

File SMY.out

sm_code	y_code	SMInv	SMEff	F_SM_smy
combatant	FY06	92	92.00	8286.84
combatant	FY07	90	90.00	11049.12
...				

The fields are SMY.sm_code, SMY.y_code, SMY.SMInv, SMInv.SMEff, SMY.F_SM_smy

106

File AMY.out

am_code	y_code	AMInv	AMEff	F_AM_amy
fighter	FY06	991	991.00	0.00
fighter	FY07	897	897.00	0.00
...				

The fields are AMY.am_code, AMY.y_code, AMY.AMInv, AMY.AMEff, AMY.F_AM_amy

File SPY.out

s_code	p_code	y_code	SPROC
DDG	Bath	FY06	0
DDG	Bath	FY07	0
DDG	Bath	FY08	0
DDG	Bath	FY09	0
DDG	Bath	FY10	0
DDG	Bath	FY11	2
DDG	Bath	FY12	2
DDG	Bath	FY13	0
...			

The fields are SPY.s_code, SPY.p_code, SPY.y_code, SPY.SPROC

File AYI.out

a_code	y_code	i	ASEG
JSFN	FY06	1	1
JSFN	FY06	2	0
JSFN	FY06	3	0
JSFN	FY06	4	0
JSFN	FY07	1	1
JSFN	FY07	2	0
JSFN	FY07	3	0
JSFN	FY07	4	0
JSFN	FY08	1	1
JSFN	FY08	2	0
JSFN	FY08	3	0
JSFN	FY08	4	0
JSFN	FY09	1	1
JSFN	FY09	2	0
JSFN	FY09	3	0
...			

The fields are AYI.a_code, AYI.y_code, AYI.i, AYI.ASEG

The Table "General" does not contain indices but contains results. It has the following features:

- Table General:
  - ➤ The associated file with results is G.out
  - ➤ It will be located in the *<path*\Results> folder
  - ➤ There is a fixed format for the first seven rows of this file:
    - ○ Row 1 may be disregarded
    - ○ Row 2 contains the following result fields:
      - ▪ Columns 1 through 12: G.Prog_Status (Integer, 12 digits)

        This code means:

General.Prog_Status Value	Meaning
1	Program executed without errors
2	An error occurred

- ▪ Columns 16 through 27: G.Sol_Status. (Integer, 12 digits).

  This code means:

General.Sol_Status Value	Meaning
1	Optimal solution
2	Feasible solution
3	Problem infeasible
4	Error while optimizing
5	Error while reading data
6	Error while initializing

- ▪ Columns 31 through 42: G.Error_Code (Integer, 12 digits) (It will be set to zero if G.Prog_Status=1).
- ▪ Columns 46 through 57: G.Error_Line (Integer, 12 digits) (It will be set to zero if G.Prog_Status=1).
  Columns 61 through 72: G.F_B (Real, 12 digits, 2 for decimal digits). (It will be set to zero if G.Prog_Status=2).
- ▪ Columns 76 through 87: G.F_CumB (Real, 12 digits, 2 for decimal digits). (It will be set to zero if G.Prog_Status=2).
- ▪ Columns 91 through 102: G.F_L (Real, 12 digits, 2 for decimal digits). (It will be set to zero if G.Prog_Status=2).
- ▪ Columns 106 through 117: G.F_SM (Real, 12 digits, 2 for decimal digits). (It will be set to zero if G.Prog_Status=2).
- ▪ Columns 121 through 132: G.F_AM (Real, 12 digits, 2 for decimal digits). (It will be set to zero if G.Prog_Status=2).
- ▪ Columns 136 through 147: G.F (Real, 12 digits, 2 for decimal digits). (It will be set to zero if G.Prog_Status=2).

108

> In addition, there is a log file "CIPA.log" located in the *<path\Results>* folder. That file contains the computational time of the different parts of the program. Also, in case of an error, it gives more details of the possible causes.

Example 1: When no error occurs General.Prog_Status=1.

G.out

Prog_Status	Sol_Status	Error_Code	Error_Line	F_B ...	1
2	0	0	5927.06		

The fields are G.Prog_Status, G.Sol_Status, G.Error_Code, G.Error_Line, G.F_B, G.F_CumB, G.F_L, G.F_SM, G.F_AM, G.F

CIPA.log

```
RESULTS FOR CASE:
 (Unspecified Code)


Program Status:                  1    (Program finished
correctly)


Solution Status:                 2    (Feasible solution)


Penalty due to Budget: F_B=                    5927.06
Penalty due to Cum. Budget: F_CumB=           12635.16
Penalty due to Labor: F_L=                    84021.90
Penalty due to Ship-Missions: F_SM=          393551.56
Penalty due to Air-Missions: F_AM=            13817.76
Total Penalty: F=                            509953.41


Time initializing parameters                    0.03
Time reading user`s data                        0.06
Time optimizing                                 4.96
    (Initial Solution)                       (  0.03)
    (SMissions)                              (  3.55)
    (AMissions)                              (  0.19)
    (Labors)                                 (  0.67)
    (SRetirements)                           (  0.11)
    (ARetirements)                           (  0.41)
Time printing results:                          0.03


Total Time CIPA:                                5.09
```

Example 2: When an error occurs General.Prog_Status=2.

G.out

Prog_Status	Sol_Status	Error_Code	Error_Line	F_B ...
2	5	471	10	0.00

The fields are G.Prog_Status, G.Sol_Status, G.Error_Code, G.Error_Line, G.F_B, G.F_CumB, G.F_L, G.F_SM, G.F_AM, G.F

CIPA.log

```
RESULTS FOR CASE:
  (Unspecified Code)

Program Status:                     2     (An error occurred)

Solution Status:                    5     (Unknown due to
errors while ...)

Error Code:                         0471
... described as:                   SMY.dat: It must be smreq
>=0
... in line (ignore if zero):       000010
... with header line:               combatant    FY14

Time initializing parameters                        0.03
Time reading user`s data                            0.03
Time optimizing                                     0.00
    (Initial Solution)                      (   0.00)
    (SMissions)                             (   0.00)
    (AMissions)                             (   0.00)
    (Labors)                                (   0.00)
    (SRetirements)                          (   0.00)
    (ARetirements)                          (   0.00)
Time printing results:                              0.00

Total Time CIPA:                                    0.06
```

Remember that in this case there will not be other result files (*.out) to read.

Appendix B: New Versions of the CIPA Solver

B.1   Introduction

The internal version of the Solver (that used for development purposes, not for official deliverables) that we consider as the starting point is the so-called Ver_25. This is the version whose characteristics, model, and algorithms have been described in this report.

When a new version of the CIPA Solver is developed, many documents may need to be updated (i.e., they will contain the information for the most updated version): CIPA General Report (this document), Optimization Model, Data Structure, Hierarchical, and Flow Diagrams, etc.

As opposed to those documents, source files (for both the heuristic and the exact solvers), executable codes, data files, and result files will be associated with specific versions. This means that they will appear under a folder with the version name on it:

```
...cipa path
       \exe
              \ver_25 (executable code "Cipa.exe" for ver_25)
              ...
              \ver_28 (executable code "Cipa.exe" for ver_28)
       \data
              \ver_25 (contains data files for cases to be run with ver_25)
              ...
              \ver_28 (contains data files for cases to be run with ver_28)
       \source
              \ver_25 (contains heuristic source files for version ver_25)
              ...
              \ver_28 (contains heuristic source files for version ver_28)
       \gams
              \ver_25 (contains exact source files for version ver_25)
              ...
              \ver_28 (contains exact source files for version ver_28)
       \results
              \ver_25 (contains result files for cases run with ver_25)
              ...
              \ver_28 (contains result files for cases run with ver_28)
```

Table B.1 describes the features in newer solver versions.

SOLVER VERSION	CHARACTERISTICS	CIPA VERSION
Ver_25 (November 13, 2001)	(Described in latest version of "Working_Report.doc") Heuristic and Exact solvers Heuristic solver features: Initial Solution Lower Bound Basic search: Mission, Labor, Budget, Retirement Deep search: Ret-Proc., Exchange (Ship, Air, Mixed, and Plant) Exact solver features: Lower bound (optional) Upper bound (optional): Initial solution (only Ship Proc. integer), Rounding (to integer and multiple of squadron size), Postsolve	P.07.04 (Delivered on November 13, 2001) Remark: The Exact Solver call is blocked out.
Ver_26 (March 14, 2002)	Increase maximum dimensions to accommodate more platforms, plants, etc. Tested against a larger set of Data under the name Steve_1 provided by N81.	P.08.05 (internal)
Ver_27 (March 20, 2002)	Described in Section 3 of this document. We incorporate effectiveness ratings for mission performance of both ship and aircraft.	P.08.27 (Delivered in May 2002)
Ver_28 (March 28, 2002)	Described in Section 4 of this document. We incorporate end-effects management.	

Table B.1. Solver versions and their characteristics.

B.2 Ver_26: Dimension

B.2.1 Introduction

This change aims to accommodate larger cases than in previous versions. In particular, we want to solve the so-called "Steve_1" case. This was created from a data set provided by N81, including new platforms and missions. Most of the inventory data have been updated according to the last EPA/TOA data.

The change was implemented in version Ver_26 (and subsequent) of the Solver.

B.2.2 Data Structure

No changes.

B.2.3 Optimization Model

No changes.

B.2.4 Heuristic Source Files

CIPA_Dim.fi: Define new maximum dimensions: Max_Y=40, Max_S=60, Max_A=40, Max_P=20, Max_SM=20, Max_AM=20, Max_I=8, Max_Q=4, Max_N=10, Max_NN=1, Max_L=10, Max_LL=1

B.2.5 GAMS Source Files

No changes.

B.3   Ver_27: Effectiveness

B.3.1 Introduction

This change aims to incorporate platform effectiveness ratings (instead of one-to-one assignments) to accomplish missions. The change was suggested by N81.

The change has been implemented in version Ver_27 (and subsequent) of the Solver.

Changes involve modifications in the Data Structure, Input and Output Data Files, the Optimization Model, the Heuristic Solver Source Files, and the Exact (GAMS) Source Files. These changes are described in the following sections.

B.3.2 Data Structure

Table "Ship-Ship-Mission"

Key	Field	Description	Type	Origin	Model	Remarks
k	s_code	Code of Ship class	A12	Dat	s	
k(H)	s	Index of Ship class	I	Cal (H)	(NA)	
k	sm_code	Code of Ship-Mission	A12	Dat	$m \in M^S$	
k(H)	sm	Index of Ship-Mission	I	Cal(H)	(NA)	
	Allowed_ssm (*)	Whether a ship class can perform a Ship-Mission or not	L	Cal (H)	S_m	'Yes' if the record exists
	SEff	Effectiveness rating	R	Dat	$seff_{sm}$	If =0, the record can be deleted

(*) The field may be omitted in the database assuming that only those existing records correspond to Allowed_ssm= 'Yes'.

113

Table "Aircraft-Air-Mission"

Key	Field	Description	Type	Origin	Model	Remarks
k	a_code	Code of Aircraft type	A12	Dat	a	$a \in a_m$
k(H)	a	Index of Aircraft type	I	Cal (H)	(NA)	$a \in a_m$
k	am_code	Code of Air-Mission	A12	Dat	$m \in M^A$	
k(H)	am	Index of Air-Mission	I	Cal(H)	(NA)	
	Allowed_aam (*)	Whether an aircraft type can perform an Air-Mission or not	L	Cal (H)	A_m	'Yes' if the record exists
	AEff	Effectiveness rating	R	Dat	$aeff_{am}$	If =0, the record can be deleted

(*) The field may be omitted in the database assuming that only those existing records correspond to Allowed_aam = 'Yes'.

Table "Ship-Mission-Year"

Key	Field	Description	Type	Origin	Model	Remarks
k	sm_code	Code of Ship-Mission	A12	Dat	$m \in M^S$	
k(H)	sm	Index of Ship-Mission	I	Cal(H)	(NA)	
k	y_code	Code of Period (year)	A12	Cal (I)	(NA)	
k(H)	y	Index of Period (year)	I	Cal(H)	y	
	smreq	Number of Ship-Missions required	I	Dat	$smreq_{my}$	
	SMInv	Number ships that can perform a Ship-Mission	I	Res	$SMInv_{my}$	Update when SInv(s,y) changes
	SMEff	Overall effectiveness for a Ship-Mission	R	Res	$SMEff_{my}$	Update when SInv(s,y) changes
	F_SM_smy	Penalty for Ship-Mission shortfall	R	Res	"Cal"	UPDATE WHEN SMEFF(SM,Y) CHANGES

Table "Air-Mission-Year"

Key	Field	Description	Type	Origin	Model	Remarks
k	am_code	Code of Air-Mission	A12	Dat	$m \in M^A$	
k(H)	am	Index of Air-Mission	I	Cal(H)	(NA)	
k	y_code	Code of Period (year)	A12	Cal (I)	(NA)	
k(H)	y	Index of Period (year)	I	Cal(H)	y	
	amreq	Number of Air-Missions required	I	Dat	$amreq_{my}$	
	AMInv	Number aircraft that can perform an Air-Mission	I	Res	$AMInv_{my}$	Update when AInv(a,y) changes
	AMEff	Overall effectiveness for an Air-Mission	R	Res	$AMEff_{my}$	Update when · AInv(a,y) changes
	F_AM_amy	Penalty for Air-Mission shortfall	R	Res	"Cal"	UPDATE WHEN AMEFF(AM,Y) CHANGES

Remark: Although the effectiveness [SSM].[SMEff] and [AAM].[AMEff] are fractional (and so will be the results [SMY].[SMEff] and [AMY].[AMEff]), we still keep integer values for the requirements [SMY].[smreq] and [AMY].[amreq].

Interface-Heuristic I/O Data Files

File SSM.dat

```
s_code        sm_code        SEff
DDG           combatant      1.00
DD21          combatant      1.00
CVX           carrier        1.00
SSN774        attack         1.00
LHX           amphibH        1.00
FFG           combatant      1.00
DD            combatant      1.00
. . .
```

The fields are SSM.s_code, SSM.sm_code, **SSM.SEff**

File AAM.dat

```
a_code        am_code        AEffect
JSFN          fighter        1.00
F18EF         fighter        1.00
F18AB         fighter        1.00
F18CD         fighter        1.00
F14           fighter        1.00
```

The fields are AAM.a_code, AAM.am_code, **AAM.AEff**

File SMY.out

sm_code	y_code	SMInv	SMEff	F_SM_smy
combatant	FY06	92	92.00	8286.84
combatant	FY07	90	90.00	11049.12
...				

The fields are SMY.sm_code, SMY.y_code, SMY.SMInv, **SMInv.SMEff**, SMY.F_SM_smy

File AMY.out

am_code	y_code	AMInv	AMEff	F_AM_amy
fighter	FY06	991	991.00	0.00
fighter	FY07	897	897.00	0.00
...				

The fields are AMY.am_code, AMY.y_code, AMY.AMInv, **AMY.AMEff**, AMY.F_AM_amy

B.3.3 Optimization Model

The following is a revision highlight of the changes in the mathematical formulation of the model to accommodate effectiveness.

SETS AND INDICES

- Mission (remain the same, we bring them here for the sake of clarity)

M^A, set of air missions; $m \in M^A$

M^S, set of ship missions; $m \in M^S$

$A_m \subseteq A$, subset of aircraft types that contribute to mission $m \in M^A$

$S_m \subseteq S$, subset of ship classes that contribute to mission $m \in M^S$

PARAMETERS (and Units)

- Constraint-related parameters: Missions

$seff_{sm}$, **effectiveness** for ship $s \in S_m$ performing mission $m \in M^S$ (# of missions per ship)

$aeff_{am}$, **effectiveness** for aircraft $a \in A_m$ performing mission $m \in M^A$ (# of missions per aircraft)

$\underline{smreq}_{my}$, **overall** effectiveness required for Ship-Mission $m \in M^S$ in time period $y \in Y$ (# missions)

$\underline{amreq}_{my}$, **overall** effectiveness required for Air-Mission $m \in M^A$ in time period $y \in Y$ (#missions)

116

DECISION VARIABLES (and Units)

- Control decision variables

$AINV_{ay}$,   inventory of type $a \in A$ aircraft at the start of year $y \in Y$ (# aircraft)

~~$AMINV_{my}$,   inventory for air mission $m \in M^A$ at the start of year $y \in Y$ (# aircraft)~~

$AMEff_{my}$,   **overall** effectiveness achieved for Air-Mission $m \in M^A$ in year $y \in Y$ (# missions)

$SINV_{sy}$,   inventory of class $s \in S$ ships at the start of year $y \in Y$ (# ships)

~~$SMINV_{my}$,   inventory for ship mission $m \in M^S$ at the start of year $y \in Y$ (# ships)~~

$SMEff_{my}$,   **overall** effectiveness achieved for Ship-Mission $m \in M^S$ in year $y \in Y$ (# missions)

FORMULATION

- Mission inventory

~~$$SMINV_{my} = \sum_{s \in S_m} SINV_{sy}, \qquad\qquad \forall m \in M^s; \forall y \in Y \qquad (11)$$~~

~~$$SMINV_{my} + \alpha_{my}^{SM} \geq smreq_{my}, \qquad\qquad \forall m \in M^s; \forall y \in Y \qquad (12)$$~~

~~$$AMINV_{my} = \sum_{a \in A_m} AINV_{ay}, \qquad\qquad \forall m \in M^A; \forall y \in Y \qquad (13)$$~~

~~$$AMINV_{my} + \alpha_{my}^{AM} \geq amreq_{my}, \qquad\qquad \forall m \in M^A; \forall y \in Y \qquad (14)$$~~

$$SMEff_{my} = \sum_{s \in S_m} seff_{sm}\, SINV_{sy}, \qquad\qquad \forall m \in M^s; \forall y \in Y \qquad \textbf{(11)}$$

$$SMEff_{my} + \alpha_{my}^{SM} \geq smreq_{my}, \qquad\qquad \forall m \in M^s; \forall y \in Y \qquad \textbf{(12)}$$

$$AMEff_{my} = \sum_{a \in A_m} aeff_{am}\, AINV_{ay}, \qquad\qquad \forall m \in M^A; \forall y \in Y \qquad \textbf{(13)}$$

$$AMEff_{my} + \alpha_{my}^{AM} \geq amreq_{my}, \qquad\qquad \forall m \in M^A; \forall y \in Y \qquad \textbf{(14)}$$

- Non-negativity and bounds

~~$$AMINV_{my} \geq 0, \qquad\qquad \forall m \in M^A; \forall y \in Y \qquad (27)$$~~

~~$$SMINV_{my} \geq 0, \qquad\qquad \forall m \in M^s; \forall y \in Y \qquad (29)$$~~

117

$$AMEff_{my} \geq 0, \qquad\qquad\qquad \forall m \in M^{A}; \forall y \in Y \qquad \textbf{(27)}$$

$$SMEff_{my} \geq 0, \qquad\qquad\qquad \forall m \in M^{s}; \forall y \in Y \qquad \textbf{(29)}$$

B.3.4 Heuristic Source Files

CIPA_DAT.fi:	Define new data: SEff(s,sm), AEff(a,am)
CIPA_RES.fi:	Define new results: SMff(sm,y), AMEff(am,y)
READ_AAM.f:	Read AEff(a,am); Check consistency AEff(a,am)≥0; Assign Aircraft-to-Mission iff AEff(a,am)>0;
READ_SSM.f:	Read SEff(s,sm); Check consistency SEff(s,sm)≥0; Assign Ship-to-Mission iff SEff(s,sm)>0;

CIPA_ERR.fi: Add data errors to the error list:
                                        Msg(287)=SSM_dat//': It must be SEff >= 0'
                                        Msg(327)=AAM_dat//': It must be AEff >= 0'

LB_AM.f:	Compute maximum possible Air-Mission effectiveness Max_AMEff and calculate the Air-Mission lower bound
LB_SM.f:	Compute maximum possible Ship-Mission effectiveness Max_SMEff and calculate the Ship-Mission lower bound
UP_AMINV.f:	Update Air-Mission effectiveness
UP_SMINV.f:	Update Ship-Mission effectiveness
UP_F_AM_amy.f:	Update cost of Air-Mission effectiveness instead of inventory
UP_F_SM_smy.f:	Update cost of Ship-Mission effectiveness instead of inventory
GAMS_AAM.f:	Add writing a new file, Par_AAM.dat containing AEff(a,am)
GAMS_SSM.f:	Add writing a new file, Par_SSM.dat containing AEff(a,am)

B.3.5 GAMS Source Files

Par_AAM.gms (New):	GAMS file to read AEff(a,am)
Par_SSM.gms (New):	GAMS file to read SEff(s,sm)

    Vars.gms: Positive decision variables AMEff(am,y) and SMEff(sm,y) substitute AMInv(am,y) and SMInv(sm,y)

Eqs.gms: Equations SMiss_1, SMiss_2, AMiss_1, AMiss_2 (eqs. (11) to (14) of the model, respectively) must be modified according to the new eqs. (11) to (14)

B.4 Ver_28: End-Effects

B.4.1. Introduction

This change aims to account for end-effects. End-effects arise especially when (a) no future missions are visualized, and (b) the cost and labor structure of some platforms impede spending money or labor for deliveries.

To overcome this problem, we incorporate the idea of "set aside budget" (for ships and aircraft) and "set aside labor" for ships. According to this, the planner may specify maximum amounts of these categories to be set aside for years beyond the plan's scope. The maximum labor to be set aside is specified by plant and year. In addition, we consider a relation between set aside labor and set aside budget for ships.

The change has been implemented in version Ver_28 (and subsequent) of the Solver.

Changes involve modifications in the Data Structure, Input and Output Data Files, the Optimization Model, the Heuristic Solver Source Files, and the Exact (GAMS) Source Files. These changes are described in the following sections.

B.4.2 Data Structure

Table "Year"

Key	Field	Description	Type	Origin	Model	Remarks
k	y_code	Code of Period (year)	A12	Cal (I)	*NA*	*Y*, set of periods, from G.Year_Ini thr G.Year_End
		...				
	max_ssab	Maximum set aside budget for ships	R	Dat	$\overline{ssab}_y$	
	max_asab	Maximum set aside budget for aircraft	R	Dat	$\overline{asab}_y$	
		...				
	SSABudget	Set aside budget for Ships	R	Res	$SSABudget_y$	Update when SALabor(p,y) changes
	ASABudget	Set aside budget for Aircraft	R	Res	$ASABudget_y$	
		...				
	Budget	Required budget	R	Res	$Budget_y$	Update when SBudget(y), ABudget(y), OMBudget(y), SSABudget(y), ASABudget(y), change

Table "Plant"

Key	Field	Description	Type	Origin	Model	Remarks
k	p_code	Code of plant	A12	Dat	p	P, set of plants
	lcrate	Labor cost rate of reference for setting aside labor and budget	R	Dat	$lcrate$	

Table "Plant-Year"

Key	Field	Description	Type	Origin	Model	Remarks
k	p_code	Code of plant	A12	Dat	p	
	max_sal	Maximum labor set aside	I	Dat	$\overline{sal}_{py}$	
	SALabor	Labor set aside	I	Res	$SALabor_{py}$	Determines $SSAB_y$
	LABOR	Required labor	I	Res	$Labor_{py}$	Update when SPROC(s,p,y), SALabor(y) changes

Interface-Heuristic I/O Data Files

File Y.dat

```
y_code      oscn        ocscn       oapn        ocapn       oom      ...
FY06        0.00        532.71      0.00        4356.30     4839.92
FY07        35.00       634.11      0.00        4874.09     4774.40
FY08        0.00        282.25      0.00        5323.42     4765.31
FY09        35.00       516.20      0.00        4721.60     4661.75
FY10        0.00        1660.92     0.00        5509.91     4669.37
FY11        35.00       391.85      0.00        6101.37     4537.59
...
```

The fields are Y.y_code, Y.oscn, Y.ocscn, Y.oapn, Y.ocapn, Y.oom, Y.toa_up, Y.toa_lo, Y.Cumtoa_up, Y.Cumtoa_lo, **Y.max_ssab**, **Y.max_asab**, Y.Alpha_BPlus, Y.Alpha_BMinus, Y.Alpha_CumBPlus, Y.Alpha_CumBMinus

File P.dat

```
p_code      lcrate      Alpha_LPlus     Alpha_LMinus
Bath        0.58        0.45            0.60
Ingals      0.60        0.22            0.29
News        0.60        0.45            0.61
Eboat       0.30        0.48            0.64
Avon        0.10        0.41            0.55
```

The fields are P.p_code, **P.lcrate**, P.Alpha_LPlus, P.Alpha_LMinus

File PY.dat

p_code	y_code	max_sal	pcap_up	pcap_lo
Bath	FY06	5000	9000	2800
Bath	FY07	5000	9000	2800
Bath	FY08	5000	9000	2800
Bath	FY09	5000	9000	2800
...				
Bath	FY25	5000	9000	2800
Ingals	FY06	5000	17000	6500
Ingals	FY07	5000	17000	6500
Ingals	FY08	5000	17000	6500
...				

The fields are PY.p_code, PY.y_code, **PY.max_sal**, PY.pcap_up, PY.pcap_lo

File Y.out

y_code	SBudget_y	SBudget	SSABudget	ABudget_y	ABudget	ASABudget ...
FY06	12962.10	13899.65	0.00	8038.74	16591.83	0.00
FY07	9967.92	10955.09	0 00	8618.67	16735.75	0.00
FY08	10845.90	11462.00	0.00	9121.92	17284.30	0.00
FY09	8499.41	9321.08	0.00	11206.09	17427.19	0.00
...						

The fields are Y.y_code, Y.SBudget_y, Y.SBudget, **Y.SSABudget**, Y.ABudget_y, Y.ABudget, **Y.ASABudget**, Y.OMSBudget_y, Y.OMABudget_y, Y.OMBudget_y, Y.OMBudget, Y.Budget, Y.CumBudget, Y.F_BPlus_y, Y.F_CumBPlus_y, Y.F_BMinus_y, Y.F_CumBMinus_y, Y.F_LPlus_y, Y.F_LMinus_y, Y.F_B_y, Y.F_CumB_y, Y.F_L_y, Y.F_SM_y, Y.F_AM_y, Y.F_y

File PY.out

p_code	y_code	SALabor	LABOR	F_LPlus_py	F_LMinus_py	F_L_py
Bath	FY06	0.00	4209	0.00	0.00	0.00
Bath	FY07	0.00	6180	0.00	0.00	0.00
Bath	FY08	0.00	7292	0.00	0.00	0.00
Bath	FY09	0.00	8174	0.00	0.00	0.00
Bath	FY10	0.00	9292	131.40	0.00	131.40
Bath	FY11	0.00	8358	0.00	0.00	0.00
...						

The fields are PY.p_code, PY.y_code, **PY.SALabor**, PY.LABOR, PY.F_LPlus_py, PY.F_LMinus_py, PY.F_L_py

B.4.3 Optimization Model

The following is a revision highlight of the changes in the mathematical formulation of the model to accommodate end-effects.

PARAMETERS (and Units)

■ Constraint-related parameters: Budget

$\overline{ssab}_y$,   maximum set aside ship budget for year $y \in Y$ ($)

$\overline{asab}_y$,   maximum set aside aircraft budget for year $y \in Y$ ($)

121

- Constraint-related parameters: Labor

$\overline{sal}_{py}$, maximum set aside labor at plant $p \in P$ in time period $y \in Y$ (# workers)

$\overline{lcrate}_p$, approximate labor cost at plant $p \in P$ for set aside labor purposes ($/worker)

DECISION VARIABLES (and Units)

- Main decision variables

$SSABudget_y$, amount of budget set aside in year $y \in Y$ for future ship procurements ($)

$ASABudget_y$, amount of budget set aside in year $y \in Y$ for future aircraft procurements ($)

$SALabor_{py}$, amount of labor set aside in year $y \in Y$ for future ship procurements from plant $p \in P$ (# workers)

FORMULATION

- Budget

$$\cancel{BUDGET_y = SBUDGET_y + ABUDGET_y + OMBUDGET_y,}$$

$$\cancel{\forall y \in Y \qquad (18)}$$

$$BUDGET_y = SBUDGET_y + ABUDGET_y + OMBUDGET_y +$$
$$SSABudget_y + ASABudget_y,$$

$$\forall y \in Y \qquad (18)$$

- Industrial

$$LABOR_{py} = clabor_{py} +$$

$$\cancel{\sum_{s \in S | p \in P_s} \sum_{\substack{y' \in Y | \\ y \le y' \le y + SCb_{sp}}} \sum_{q \in Q_{spy'}} sworkb_{spq,y'-y} \; SPROC_{spy'q} +}$$

$$\cancel{\sum_{s \in S | p \in P_s} \sum_{\substack{y' \in Y | \\ y - SCa_{sp} \le y' \le y-1}} \sum_{q \in Q_{spy'}} sworka_{spq,y-y'} \; SPROC_{spy'q},}$$

$$\cancel{\forall p \in P; \forall y \in Y \qquad (23)}$$

$$LABOR_{py} = clabor_{py} + SALabor_{py} +$$

$$\sum_{s \in S | p \in P_s} \sum_{\substack{y' \in Y| \\ y \le y' \le y + SCb_{sp}}} \sum_{q \in Q_{spy'}} sworkb_{spq,y'-y} \, SPROC_{spy'q} +$$

$$\sum_{s \in S | p \in P_s} \sum_{\substack{y' \in Y| \\ y - SCa_{sp} \le y' \le y-1}} \sum_{q \in Q_{spy'}} sworka_{spq,y-y'} \, SPROC_{spy'q},$$

$$\forall p \in P; \forall y \in Y \qquad (23)$$

(New)

$$\sum_{p \in P} lcrate_p \, SALabor_{py} = SSABudget_y, \qquad \forall p \in P; \forall y \in Y \qquad (26)$$

- Non-negativity and bounds

(New)

$$0 \le SSABudget_y \le \overline{ssab}_y, \qquad \forall y \in Y \qquad (27)$$

$$0 \le ASABudget_y \le \overline{asab}_y, \qquad \forall y \in Y \qquad (28)$$

$$0 \le SALabor_{py} \le \overline{sal}_{py}, \qquad \forall p \in P; \forall y \in Y \qquad (29)$$

Note: Former constraints (26) to (46) are now indexed as (30) to (50).

B.4.4 Heuristic Source Files

CIPA_DAT.fi:	Define new data: max_ssab(y), max_asab(y), max_sal(p,y), lcrate(p)
CIPA_RES.fi:	Define new results: SSABudget(y), ASABudget(y), SSALabor(p,y)
READ_Y.f:	Read max_ssab(y), max_asab(y); Check consistency max_ssab(y)≥0.0, max_asab(y) ≥0.0
READ_P.f:	Read lcrate(p); Check consistency lcrate(p)≥0.0
READ_PY.f:	Read max_sal(p,y); Check consistency max_sal(p,y)≥0
CIPA_ERR.fi:	Add text to existing error list to for new data validations
LB_LABOR.f:	Compute maximum labor by also adding the maximum set aside labor (including budget constraints)
INIT_SHIPS:	Initialize SALabor(p,y)=0, and therefore SSABudget(y)=0.0
INIT_AIR:	Initialize ASABudget(y)=0.0
UP_BUDGET.f:	Update the budget formula using new equation (18)
UP_LABOR.f:	Update the labor formula using new equation (23)
FEAS_SSA:	(New procedure). Check for set aside labor feasibility and for ship set aside budget feasibility, according to equations (26), (27), and (29)

FEAS_ASA:	(New procedure). Check for aircraft set aside budget feasibility, according to equation (28)
FEAS_ALL:	Needs to call on FEAS_SSA and FEAS_ASA (in addition to previous feasibility procedures) in order to validate any given solution

SET_ASIDE.f:	To manage the set aside strategies
SET_ASIDE_INI:	To calculate a reasonable incremental rate to determine new set aside options
SET_ASIDE_SHIP:	To analyze a variety of new set aside options for labor and ship budget
SET_ASIDE_AIR:	To analyze a variety of new set aside options for aircraft budget
OPTIMIZE:	To call on the SET_ASIDE procedure (in addition to the others local search strategies)

GAMS_Y.f:	Add writing parameters max_ssab(y), max_asab(y) to file Par_Y.dat used by GAMS
GAMS_P.f:	Add writing parameter lcrate(p) to file Par_P.dat used by GAMS
GAMS_PY.f:	Add writing parameter max_sal(p,y) to file Par_PY.dat used by GAMS
READ_GAMS_Y.f:	Add reading results SSABudget(y), ASABudget(y) from file \gams\results\Y.out after GAMS optimization
READ_GAMS_PY.f:	Add reading results SALabor(p,y), from file \gams\results\PY.out after GAMS optimization

WRITE_G.f:	Add printing the timing of the SET_ASIDE procedure to G.out
WRITE_Y.f:	Add printing the solution SSABudget(y), ASABudget(y) to Y.out
WRITE_PY.f:	Add printing the solution SSALabor(p,y) to PY.out

B.4.5 GAMS Source Files

Par_Y.gms:	Read also max_ssab(y), max_asab(y)
Par_P.gms:	Read also lcrate(p)
Par_PY.gms:	Read also max_sal(p,y)

Vars.gms:	Positive decision variables SSABudget(y), ASABudget(y), SSALabor(p,y)

Eqs.gms:	Add equation Labor_4 to represent the new constraint (26). Modify constraints Budget_1 and Labor_1 according to the new equations (18) and (23), respectively.
Bou_vars.gms:	Add bounds for new variables, equations (27)-(29)

Round_Parameters.gms:	Add parameter r_SALabor(p,y) for rounding SALabor(p,y) variable to an integer value
Round_Fix.gms:	Round SALabor(p,y) to its integer part and fix it before resolving

Out_y.gms: Print   SSABudget(y)   and   ASABudget(y)   solution   to
                    \gams\results\Y.out
Out_py.gms: Print SALabor(p,y) solution to \gams\results\PY.out

References

Baker, S., [2001], "Unmanned Air Combat Vehicle 2010," *Air Force Materiel Command News Service*, September 24.

Baran, N., [2000], "Optimizing Procurement Planning of Navy Ships and Aircraft," MS Thesis in Operations Research, Naval Postgraduate School, Monterey, CA, December.

Boeing [2001], An artist's rendition of the Boeing Unmanned Combat Air Vehicle (UCAV). [http://www.Boeing.com], accessed August.

Brooke, A., Kendrick, D., Meeraus, A., and Raman, R., [1998], "GAMS User's Guide," GAMS Development Corporation, Washington, DC.

Capaccio, T., [2001], "Emergency Defense Funds Going To Improve Military Intelligence," [http://www.Bloomberg.com], accessed September 27.

Chief of Naval Operations, [2000], "Vision..Presence..Power-A Program Guide to the U.S. Navy, 2000 Edition," [http://www.chinfo.navy.mil/navpalib/policy/vision/Vis00/v00-ch2b.html], accessed March 2002.

CIPA: Data Structures [2002], Naval Postgraduate School (development document).

CIPA: Quick Tour [2001], Naval Postgraduate School (PowerPoint presentation).

CIPA Solver Versions [2002], Naval Postgraduate School (development document).

CIPA: User's Manual [2001], Naval Postgraduate School (in preparation).

Digital Visual Fortran [1998], Professional Edition 6.0.0., Digital Equipment Corporation.

Field, R., [1999], "Planning Capital Investment in Navy Forces," MS Thesis in Operations Research, Naval Postgraduate School, Monterey, CA, December.

GAMS/CPLEX, [2002], [http://www.gams.com/solvers/solvers.htm#CPLEX], accessed February 7.

GAMS/OSL [2002], [http://www.gams.com/solvers/solvers.htm#OSL], accessed February 7.

Garcia, R. M., [2001], "Optimized Procurement and Retirement Planning of Navy Ships and Aircraft," MS Thesis in Operations Research, Naval Postgraduate School, Monterey, CA, December.

Grossman, E. M., [2001], "Rumsfeld Changes Yardstick For Measuring And Shaping U.S. Forces," *Inside The Pentagon*, July 26.

Hagan, K. J., [1978], *In Peace and War: Interpretations of American Naval History, 1775-1978*, Greenwood Press, Westpoint, Connecticut.

ILOG [2002], "ILOG CPLEX," [http://www.ilog.com/products/cplex/], accessed February 7.

Lockheed Martin, [2001], An artist's rendition of the Lockheed Martin Joint Strike Fighter (JSF).

Martin, L.W., [1988], "Economics of War," appears in Encyclopedia Britannica, ed. Britannica Editors, Encyclopaedia Britannica, Inc.

Microsoft Excel [2002], Microsoft Corporation, [http://www.microsoft.com/office/excel/default.htm], accessed February 7.

Navy Public Affairs Library, [1997], "F/A-18 E/F Super Hornet," The Navy Office of Information, http://www.chinfo.navy.mil/navpalib/, accessed August 2001.

Navy Public Affairs Library, [2000], "At Least Six Sailors Killed in Small Boat Attack on USS Cole," The Navy Office of Information, October 1.

Pasztor, A., Squeo, A. M., and Rogers, D., [2001], "Boeing Could Be First to Benefit From Growing Defense Spending," *Wall Street Journal*, October 2.

OSL [2002], "IBM OSL," [http://www-4.ibm.com/software/data/bi/osl/], accessed February 7.

Rhem, K., [2001], "Bush: No Distinction Between Attackers and Those Who Harbor Them," *American Forces Press Service*, September 11.

Scarborough, R., [2001a], "Bush's 'Strategy First' Vow Scrapped, Review Officials Say," *Washington Times*, July 23.

Scarborough, R., [2001b], "Defense Review To Feel Impact Of Terror Attacks," *Washington Times*, September 27.

Thomas, C., [2001], "Rumsfeld's New Defense Strategies," *Washington Times*, August 1.

System Planning and Analysis (SPA), [2000], "SPA Briefing to Naval Center for Cost Analysis—SPA Extended Planning Annex Model," briefing presented to Naval Center for Cost Analysis, Washington, DC, March 16.

USS CONSTITUTION, [1999], "Old Ironsides History Page,"
     [http://www.ussconstitution.navy.mil/Shiphistoryx.thm], December.

Valentine, W., [1999], "The IWAR/CPAM Assessment Process," briefing presented to
     the Naval Postgraduate School, Monterey, CA, March 18.

Weinberger, S., [2001], "Wolfowitz Promises Paradigm Shift With QDR, But Offers Few
     Details," *Aerospace Daily*, August 9.

Whittle, R., [2001], "Defense Bill May Force V-22 Production Cuts," *Dallas Morning
     News*, September 8.

Windows 2000 [2002], Microsoft Corporation,
     [http://www.microsoft.com/windows2000/], accessed February 7.

Initial Distribution List

1. Research Office (Code 09)..1
   Naval Postgraduate School
   Monterey, CA 93943-5000

2. Dudley Knox Library (Code 013)...2
   Naval Postgraduate School
   Monterey, CA 93943-5002

3. Defense Technical Information Center...2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

4. Richard Mastowski (Editorial Assistant)..2
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

5. Research Assistant Professor Javier Salmeron..3
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

6. Associate Professor Robert F. Dell...3
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

7. Distinguished Professor Gerald G. Brown..3
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

8. Research Associate Anton Rowe...3
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5000

9. Chief of Naval Operations (N81)..1
   ATTN: CDR Steve Richter
   2000 Navy Pentagon
   Washington, DC 20350-2000